

Visual Analysis of Anatomy Ontologies and Related Genomic Information

Aba-Sah Dadzie



Degree of Doctor of Philosophy

Heriot-Watt University Department of Computer Science

July 2006

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that the copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author or of the University (as may be appropriate).

Abstract

Challenges in scientific research include the difficulty in obtaining overviews of the large amount of data required for analysis, and in resolving the differences in terminology used to store and interpret information in multiple, independently created data sets. Ontologies provide one solution for analysis involving multiple data sources, improving cross-referencing and data integration.

This thesis looks at harnessing advanced human perception to reduce the cognitive load in the analysis of the multiple, complex data sets the bioinformatics user group studied use in research, taking advantage also of users' domain knowledge, to build mental models of data that map to its underlying structure. Guided by a user-centred approach, prototypes were developed to provide a visual method for exploring users' information requirements and to identify solutions for these requirements. 2D and 3D node-link graphs were built to visualise the hierarchically structured ontology data, to improve analysis of individual and comparison of multiple data sets, by providing overviews of the data, followed by techniques for detailed analysis of regions of interest.

Iterative, heuristic and structured user evaluations were used to assess and refine the options developed for the presentation and analysis of the ontology data. The evaluation results confirmed the advantages that visualisation provides over text-based analysis, and also highlighted the advantages of each of 2D and 3D for visual data analysis.

Acknowledgements

My thanks go first to my supervisor, Albert Burger, for his guidance throughout the process of completing my research and thesis.

I am very grateful also to Gus Ferguson, for the time he spent reviewing my work, especially while I was preparing to carry out the evaluations that made a significant contribution to my research, for the important and useful suggestions he made.

Thanks to Jeff Christiansen at Edinburgh's Medical Research Council, who took a lot of time out of his schedule to help me prepare for my first structured evaluation. And to Lorna and Venkat who also spent quite a bit of time reviewing the visualisation tool I developed as part of my work, in between my two major evaluations.

I must also thank the researchers at the MRC and at Heriot-Watt's School of MACS who took part in the several evaluations of my visualisation browsers, for the insightful comments each made on usability and usefulness of the visualisation techniques I explored as part of my research. Unfortunately for ethical reasons I cannot name these people; this however does not detract from the appreciation I owe to each one.

I think in pictures — my sincerest thanks to each person who gave me permission to make use of diagrams in work they have published in information visualisation, to illustrate concepts I discuss in this thesis. I have learnt a lot from the contribution each of you has made to a field that communicates knowledge in ways words are unable to.

I also wish to acknowledge Universities UK, who used to administer the Overseas Research Students Awards Scheme, which provided funding for the first three years of my PhD and qualified me also for the Heriot-Watt University James Watt Scholarship.

Finally there is a group of people who fall under the criterion *etc* - each of you knows who you are, and what support you gave me. I thank each and every one of you. This is to all who believed in me more than I did myself.

ACADEMIC REGISTRY
Research Thesis Submission



Name:			
School/PGI:			
Version: <i>(i.e. First, Resubmission, Final)</i>		Degree Sought:	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

- 1) the thesis embodies the results of my own work and has been composed by myself
- 2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
- 3) the thesis is the correct version of the thesis for submission*.
- 4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying, subject to such conditions as the Librarian may require
- 5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	
-------------------------	--	-------	--

Submission

Submitted By <i>(name in capitals)</i> :	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in Academic Registry

Received in the Academic Registry by <i>(name in capitals)</i> :			
<i>Method of Submission</i> <i>(Handed in to Academic Registry; posted through internal/external mail):</i>			
Signature:		Date:	

Contents

List of Figures	ix
List of Abbreviations	xiii
List of Publications	xvi
1 Introduction	1
1.1 Data analysis	1
1.1.1 Visual data analysis	2
1.1.2 Bioinformatics and biological data analysis	3
1.1.3 Ontologies in data analysis	3
1.2 The Edinburgh Mouse Atlas Project	4
1.3 The Cross-Species Anatomy Network	4
1.4 Structure of thesis	4
1.4.1 Identifying issues pertinent to visual data analysis	4
1.4.2 Involving the user in developing a solution	5
1.4.3 Research findings	7
2 Data analysis and information processing in humans	9
2.1 Information processing in humans	9
2.2 Data analysis	9
2.2.1 The process of data analysis	10
2.2.2 Collaborative work and incremental data analysis	11
2.3 Visual data analysis	12
2.3.1 Information visualisation	12
2.3.2 Metaphors in visualisation	13
2.3.3 Navigation through data	17
2.3.4 Issues in visual data analysis	18
2.4 Browsing, searching and querying data	21
2.5 Visualisation in the development of data analysis tools	23
2.5.1 Importance of a modular approach to development	24
2.5.2 Existing techniques for data analysis	24

2.5.3	Interactivity and animation	35
2.5.4	Limitations in visual analysis	36
2.6	2D vs 3D	36
2.7	Analysing data over a network	39
2.8	Summary	39
3	Graph visualisation	41
3.1	Common applications of graph visualisation	41
3.2	A review of existing graph visualisation tools	42
3.3	Limitations in tree graph visualisation	47
3.4	Summary	48
4	Bioinformatics data analysis	49
4.1	Ontologies in bioinformatics	49
4.1.1	Anatomy Ontologies	51
4.2	Tools and techniques for bioinformatics data analysis	52
4.2.1	Bio-ontology databases and tools	55
4.3	Applications of bioinformatics	62
4.4	Ethical issues in bioinformatics research	63
4.5	Summary	63
5	Challenges in the analysis of anatomy ontologies	65
5.1	The Edinburgh Mouse Atlas Project	66
5.1.1	Structure of EMAP anatomy ontology	67
5.1.2	Access and interoperability	69
5.2	The Cross-Species Anatomy Network	69
5.2.1	Access and interoperability	71
5.3	Data analysis requirements for EMAP and XSPAN	72
5.3.1	Recognition of data structure	72
5.3.2	Alternative structuring of data	72
5.3.3	Tracing lineage within data	73
5.3.4	Representation of complex relationships	73
5.3.5	Visual, dynamic querying	75
5.3.6	Multiple, simultaneous analysis of ontologies	77
5.4	Graphical analysis of bio-ontologies	78
5.4.1	Assessment of existing graphical analysis techniques	78
5.5	Proposal for a solution for data analysis	81
5.6	Summary	83

6	Developing solutions employing visual analysis	85
6.1	Choice of programming language	86
6.2	Data types and storage methods	86
6.2.1	Database	87
6.2.2	XML	87
6.2.3	Image	87
6.3	Structure of application	87
6.3.1	Designing for modularity and extensibility	87
6.3.2	The data access layer	87
6.3.3	The visualisation layer	90
6.4	Practical considerations in layout of node-link graphs	90
6.5	The 2D browser	91
6.5.1	Design of visualisation graphs	91
6.5.2	Browser design	93
6.5.3	Heuristic evaluation of the initial prototype for the 2D browser	94
6.5.4	Options provided for analysis in 2D	96
6.5.5	Limitations in the 2D browser	106
6.6	Visualisation in 3D? Resolving limitations in 2D	106
6.7	The 3D browser	107
6.7.1	Choice of programming language	107
6.7.2	Design of visualisation graphs	108
6.7.3	Browser design	109
6.7.4	Encoding of data properties	110
6.7.5	Overcoming limitations to analysis in 2D	111
6.7.6	Further options for analysis in 3D	113
6.7.7	Limitations in the 3D browser	114
6.8	Related work in the field	115
6.9	Summary	117
7	Structured usability evaluation of visualisation prototypes	118
7.1	Preparation for usability evaluation	118
7.1.1	Test hypotheses	119
7.1.2	Preparation of evaluation documents	119
7.1.3	Pilot test and expert review of evaluation procedure	122
7.2	Assessment of variation in system response in 2D	122
7.3	Implementation of evaluation procedure	125
7.3.1	User backgrounds	125
7.3.2	Evaluation procedure	128
7.4	Analysis of evaluation results	130
7.4.1	SUS Scores	130

7.4.2	General satisfaction ratings	130
7.4.3	Assessment of the 2D browser	132
7.4.4	Assessment of the 3D browser	134
7.4.5	Task completion times	134
7.5	Discussion of evaluation findings	138
8	Visual solutions for data analysis	140
8.1	Changes to prototypes based on evaluation results	140
8.1.1	Graph attributes and layout	140
8.1.2	Supplementary textual detail	141
8.1.3	Editing data structure	142
8.1.4	Search options	143
8.1.5	Navigation and exploration	144
8.2	Additional suggestions for changes to browsers	144
8.3	Solutions for open analysis issues	146
8.3.1	Querying with a direct-manipulation interface	146
8.3.2	Mapping equivalence across multiple ontologies	147
8.3.3	Tracing lineage within and across ontologies	148
8.4	Assessment of visualisation solutions	149
9	Final evaluation of visual analysis solutions	151
9.1	Major questions addressed	151
9.1.1	Perceptual cues provided	151
9.1.2	Comparison of the browsers	153
9.1.3	Additional hypothesis	153
9.2	Evaluation design	153
9.2.1	Preparation of evaluation documents	153
9.2.2	Test run of evaluation procedure	154
9.3	Implementation of evaluation procedure	154
9.3.1	User backgrounds	154
9.3.2	Evaluation procedure	154
9.4	Analysis of results	155
9.4.1	Task completion times	156
9.4.2	SUS Scores	156
9.4.3	General satisfaction ratings	157
9.4.4	Assessment of the 2D browser	157
9.4.5	Assessment of the 3D browser	158
9.4.6	Comparison between the 2D and 3D browsers	158
9.4.7	Comparison of the visualisation browsers to the EMAP indices	160
9.4.8	Spatial ability exercises	162

9.5	Discussion of evaluation findings	163
9.5.1	Understanding of data structure	163
9.5.2	Search and query	164
9.5.3	Managing occlusion	165
9.5.4	Perceptual cues	166
9.5.5	Spatial awareness/ability	167
9.6	Review of visualisation browsers	169
9.6.1	Limitations of approach	170
9.7	Summary	171
10	Conclusions	172
10.1	Review of thesis	172
10.1.1	Identification of problem area	172
10.1.2	Development of a visual analysis solution	173
10.2	Main findings	174
10.2.1	Contribution to research	175
10.3	Conclusions	175
10.4	Future work	177
10.4.1	Extending visual analysis solutions developed	177
10.4.2	Potential solutions to performance limitations in Java	178
10.4.3	Study of factors with subjective influence on visual analysis	178
A	Sample input files	179
A.1	DTD for EMAP anatomy ontologies	179
A.2	DTD for user session XML files	180
A.3	Reloadable session file for TS11	182
B	Design documents	184
B.1	Application menu for 2D browser	184
B.2	Popup menus for 2D browser	187
B.3	Zoom pane popup menus	190
B.4	Application menu for 3D browser	191
B.5	Toolbars for 2D and 3D browsers	193
B.6	Navigation aids for the 3D browser	194
B.6.1	Actions associated with <i>MouseBehaviors</i>	194
B.6.2	Actions associated with <i>KeyNavigatorBehaviors</i>	194
C	Evaluation documents	195
C.1	User instruction sheet	196
C.2	Task scenario sheets	197
C.3	Questionnaires	202

C.3.1	Pre-evaluation questionnaire	202
C.3.2	Post-evaluation questionnaire	206
D	Evaluation results	215
D.1	Pre-evaluation questionnaire	215
D.1.1	Use of input/output devices	215
D.2	Post-evaluation questionnaire	216
D.2.1	Mean rankings over all users for each item for 2D browser	216
D.2.2	Mean rankings over all users for each item for 3D browser	217
D.3	Task completion times	218
D.3.1	Mean task completion times	218
D.3.2	Task completion times for each user	218
E	Documents for final evaluation	220
E.1	User Instruction Sheet	221
E.2	Quick guide to visualisation browsers	222
E.3	Task scenario sheets	223
E.4	Post-evaluation questionnaire	226
E.5	Spatial ability/awareness exercises	239
E.6	Sample log recording use of browsers	243
F	Results for final evaluation	244
F.1	Range of specifications for users' computers	244
F.2	Post-evaluation questionnaire	245
F.2.1	Mean usability satisfaction rankings	245
	References	246

List of Figures

2.1	A cityscape used to provide intuitive visual analysis and navigation	15
2.2	[77] uses an organic metaphor to visualise use of a web site	15
2.3	[40] demonstrate the importance of effective visual cues for interpreting visualisations	18
2.4	[48] highlight ROIs within the context of the overview using non-uniform scaling	20
2.5	[98] provide visual cues that highlight ROIs in a data set	20
2.6	Extracting an ROI from the overview, to allow detailed analysis in isolation .	21
2.7	Suppressing non-relevant data during analysis of a selected ROI	21
2.8	The structure of a document group on the web is shown using a 3D force-based layout	25
2.9	[112] illustrate the advantages tree maps provide over other hierarchical visualisation techniques	26
2.10	Removing nesting in a tree map to provide more space for displaying data . .	27
2.11	The information cube visualisation technique developed by [147]	28
2.12	A tree graph that provides additional visual cues for highlighting data of interest.	28
2.13	A cladogram generated using <i>Phylodendron</i> to visualise evolutionary divergence between seven mammals	29
2.14	Comparison of a scatter plot based on Sammon's NLM to the equivalent tree graph	29
2.15	(3D) cone and cam trees	30
2.16	[151] use a perspective wall to visualise the structure of a file system	32
2.17	[107] demonstrate the use of parallel coordinates for visualisation of multi-dimensional data	33
2.18	[69] illustrates use of the parallel co-ordinates visualisation technique in 3D .	33
2.19	Using the <i>City'O'Scope</i> visual analysis tool to analyse multi-dimensional economic data	34
3.1	Hyperlinks between documents in a web site displayed using a node-link graph	42
3.2	Structured navigation through a web site illustrated with hierarchical graphs	42
3.3	Laying out an EMAP XML file using <i>SpaceTree</i>	43
3.4	The <i>HyperGraph</i> browser is used for navigation through a wiki	44

3.5	A sample data set used to generate a 3D node-link graph using <i>VRMLgraph</i> .	45
3.6	<i>Walrus</i> used to visualise the directory structure of the folder containing its source files	45
3.7	Visualisation of phylogenetic trees using the <i>ATV Viewer</i>	46
3.8	Visualisation of relationships within biological data using <i>BioLayout JAVA</i> .	47
4.1	A phylogenetic tree drawn using <i>DrawTree</i>	54
4.2	<i>J-Express Pro</i> being used to visualise biological data	55
4.3	Alternative methods for displaying data retrieved from GO using <i>AmiGO</i> . .	56
4.4	The <i>Graph Widget</i> in <i>Protégé</i>	59
4.5	A demonstration of the <i>SHriMP</i> visualisation technique	60
4.6	<i>TouchGraph</i> used to visualise the Free Online Dictionary of Computing . . .	61
4.7	A demonstration version of <i>OntoRama</i> used to visualise ontology data	61
5.1	Online version of the EMAP section browser	67
5.2	Downloadable EMAP browser providing advanced search capability for gene expression data mapped to anatomy ontologies	68
5.3	Revealing the hierarchical structure of TS04 using a node-link graph	69
5.4	Identification of similarity in anatomical components in the <i>mouse</i> and <i>Drosophila</i>	70
5.5	Integration of the different data sources feeding into the XSPAN system . . .	71
5.6	Structure of the XSPAN prototype	72
5.7	Comparison of the lengths of the text indices representing TS04 and TS26 . .	73
5.8	Potential for alternative structuring of TS20, to create a <i>group</i> node for the component <i>skeleton</i>	74
5.9	An illustration of the method currently used to trace lineage in EMAP	74
5.10	Multiple paths to the root from a single node occurring as a result of the creation of a <i>group</i> node	75
5.11	Identifying initial and final stages of occurrence of three components during the development of the mouse embryo	76
5.12	Multiple occurrence of components in the <i>abstract mouse</i> , due to repeated entries in multiple stages of development.	76
5.13	Inferring lineage in anatomy ontologies	77
6.1	Empirical, user-centred cycle followed in development and assessment of novel approaches to visual analysis	86
6.2	Extract from the EMAP XML file for TS11	88
6.3	Comparison between current methods for storing and accessing data in EMAP and proposals for new visualisation system	89
6.4	Input data flow for the visualisation browsers	89
6.5	Structure of the visualisation layer	90

6.6	Graphical overview of the anatomy ontology for TS11 in 2D	90
6.7	The vertical layout for the DAG representing TS11	91
6.8	Radial layout for TS11, showing the first four levels in the DAG	92
6.9	Comparison of a <i>relative</i> to a <i>uniform</i> layout of nodes for TS16	93
6.10	The horizontal layout for TS11 at start-up, showing 3 levels in the graph . . .	93
6.11	Abstraction used to improve usability of the overview graph drawn for TS26, which contains 1749 nodes.	94
6.12	The visualisation application developed to hold the individual graphs drawn in 2D to represent anatomy ontologies.	95
6.13	Functions available from the toolbar in the 2D browser	95
6.14	Component detail brought up for the node <i>embryo.branchial arch</i> in the graphical representation for TS12	97
6.15	Suppression of data of lower importance, to highlight data of current interest	98
6.16	Simultaneous editing of multiple data nodes in the 2D browser	99
6.17	One implementation of zoom that redraws ROIs at higher magnification in a coupled sub-window	100
6.18	Providing a physical and a semantic zoom for an ROI in TS16, within the context of the overview	102
6.19	Physical zoom that magnifies the entire graph	103
6.20	Sub-string searching using a custom search dialog	104
6.21	Highlighting paths within a single ontology from three nodes of interest . . .	105
6.22	Use of the visualisations generated to aid creation of a <i>group</i>	105
6.23	Multiple ontologies loaded into the 3D window	108
6.24	Moving the viewpoint away from the centre of the visualisation to obtain an overview of data	109
6.25	Functions available from the toolbar in the 3D browser	110
6.26	Editable legend displaying default values for colours used to encode attributes of objects in the scene	110
6.27	Tracing lineage in the 3D browser using links between nodes in successive stages of development	111
6.28	Dialog used to record equivalence relationships between components in dif- ferent ontologies	112
6.29	Comparison between graphical representations of grouping in 2D and 3D . . .	113
6.30	Displaying textual detail for a user-created link drawn between two ontologies	114
6.31	Nodes that match search criteria are highlighted in green in the 3D window .	115
6.32	Layered di-graphs in 3D space that provide an extension to the <i>Sugiyama</i> <i>method</i>	116
7.1	Extract from the task scenario sheet for the 2D browser for the structured user evaluation	120

7.2	Comparison of system response time with data load	123
7.3	Comparison of system response time with data load after improving layout .	124
7.4	User backgrounds, recording education and current work	125
7.5	Computing skill and educational background of each user	126
7.6	Distribution of Internet connection types and speeds available to users	127
7.7	Use of web browsers	127
7.8	Prior experience using EMAP browsers	128
7.9	Comparison of mean task completion times with those for users 04 and 05 . .	129
7.10	SUS score for each user	130
7.11	Overall user satisfaction rankings for each of the visualisation browsers	131
7.12	Comparison between overall user satisfaction rankings for the two major tar- get groups	131
7.13	Mean completion time for each task for the 2D browser, compared to MTC .	135
7.14	Mean completion time for each task for the 3D browser, compared to MTC .	135
7.15	Individual task completion times for both browsers, compared to MTC	135
7.16	Comparison of task completion times for tasks T8-2D and T3-3D	137
8.1	Improved implementation of ghosting	141
8.2	The custom dialog used to create/edit <i>group</i> nodes	143
8.3	Viewing a user-created <i>group</i> in isolation	143
8.4	Additions to options for retrieving results in search dialog	144
8.5	A custom dialog is used to retrieve relationships defined between component pairs across different ontologies	147
8.6	Alternative encoding to highlight relationships across data sets	148
8.7	Auto-retrieval and tracing of lineage from a node of interest in the 3D browser	149
9.1	Frequency and length of use of the working EMAP browsers	155
9.2	SUS score for each user	156
9.3	Overall mean satisfaction ratings	157
9.4	User rankings for ability to make use of 2D browser	158
9.5	User rankings for ability to make use of 3D browser	158
9.6	Comparison of ability to use 2D and 3D browsers	159
9.7	Comparison of the 2D to the 3D browser shows a preference for the 3D . . .	160
9.8	Comparison of the EMAP text indices to the visualisation browsers	161
9.9	Comparing tracing of lineage between the text indices and the 2D and the 3D visualisation browsers	161
9.10	A user's impression of the difference in the structures created in 2D and 3D .	162
9.11	A user's impression of the difference between <i>grouping</i> in 2D and 3D	162
9.12	A user's impression of the 3D structure	162
9.13	Measurements of usefulness of options provided for detailed analysis of ROIs	165

List of Abbreviations

Abbreviation	Full Form
2D	two-dimensional
3D	three-dimensional
AI	Artificial Intelligence
BLAST	Basic Local Alignment Search Tool
CI	Confidence Interval
CLI	Command-Line Interface
CORBA	Common Object Request Broker Architecture
CRM	(GALEN) Common Reference Model
CS	Computer Science
DAG	Directed, Acyclic Graph
DFD	Data Flow Diagram
DTD	Document Type Definition
DMI	Direct Manipulation Interface
EBI	European Bioinformatics Institute
ELSI	Ethical, Legal, and Social Implications (Program)
EMAGE	The Edinburgh Mouse Atlas Gene Expression (Database)
EMAP	(The) Edinburgh Mouse Atlas Project
EMBL	European Molecular Biology Laboratory
ER	Entity Relationship (Diagram)
FMA	The Foundational Model of Anatomy
F+C	Focus+Context
GALEN	Generalised Architecture for Languages, Encyclopaedias and Nomenclatures in Medicine
GO	Gene Ontology
GOA	GO Annotation
GRAIL	The GALEN Representation and Integration Language
GUI	Graphical User Interface
GXD	Gene Expression Database (Jackson Laboratory)
HCI	Human-Computer Interaction

HCIL	Human-Computer Interaction Laboratory, University of Maryland
HTML	HyperText Markup Language
HWU	Heriot-Watt University
ID	identifier
IR	Information Retrieval
JPEG	Joint Photographic Experts Group (image file)
JVM	Java Virtual Machine
KDD	Knowledge Discovery In Databases
LR	Left-right
MACS	School of Mathematical and Computer Sciences (HWU)
MGI	Mouse Genome Informatics
MRC HGU	Medical Research Council Human Genetics Unit (Edinburgh)
MS	Microsoft
MTC	Maximum Time to Completion
N/A	Not Applicable
NLM	(Sammon's) Non-Linear Mapping
OBO	Open Biomedical Ontologies
OLAP	On-Line Analytical Processing
OO	Object-Oriented
O/S	Operating System
OWL	Web Ontology Language
PC	Personal Computer
PCA	Principal Component Analysis
QUIS	Questionnaire for User Interface Satisfaction
RAM	Random Access Memory
ROI	Region of Interest
SAEL	SOFG Anatomy Entry List
SHriMP	(the) Simple Hierarchical Multi-Perspective (technique)
SOFG	Standards and Ontologies for Functional Genomics
SD	Standard Deviations
STD	State Transition Diagram
SUS	System Usability Scale
TAMBIS	Transparent Access to Multiple Bioinformatics Information Sources Project
TaO	TAMBIS Ontology
TD	Top-down
TS	Theiler Stage
UniProt	Universal Protein Resource
VR	Virtual Reality
VRML	Virtual Reality Modelling Language

XML	eXtensible Mark-up Language
XSPAN	(The) Cross-Species Anatomy Network

List of Publications

Papers

- Dadzie, A.-S. & Burger, A. (2005).** Providing visualisation support for the analysis of anatomy ontology data, *BMC Bioinformatics* 6: 74, (21 pps).
- Dadzie, A.-S. & Burger, A. (2004).** The merits of the third dimension for visual analysis of multiple anatomy ontologies, in M. He, G. Narasimhan & S. Petoukhov (eds), *Advances In Bioinformatics And Its Applications: Proceedings of the International Conference on Bioinformatics and Its Applications (ICBA04)*, pp.576-587, World Scientific Publishing Company.

Abstracts & posters

- Dadzie, A.-S. & Burger, A. (2006).** Abstract and poster: Harnessing spatial memory to aid complex data analysis, *2006 London Hopper Colloquium*
- Dadzie, A.-S. & Burger, A. (2004).** Abstract and poster: Providing visualisation support for analysis in EMAP, *Young Bioinformaticians Forum 2004 (YBF 2004)*. (last viewed Jul 2006). Available online: http://www.ybf.org/cgi-bin/abstract_2004.cgi?abstract=Dadzie.

Chapter 1

Introduction

Improvements in technology have led to increasingly complex experiments in scientific research, resulting in a large amount of equally complex, multi-dimensional data [105, 124, 160]. The data stored, in and of itself, is not very useful; it is its information content that is valuable to researchers [137]. The same technology must be used to provide effective methods for the storage, management and analysis of the complex data it generates [151, 163, 165, 184, 185]; data generation has however exceeded the ability to manage and analyse it [129, 88].

Storing data in digital format allows a wide range of data analysis techniques to be used in information retrieval (IR). Fully-automated indexing and searching in addition to computer-aided data analysis result in more efficient and effective analysis; computers can be used to perform large amounts of repetitive processing on complex, structured data, relieving humans of what would be boring, tedious work [2]. This allows researchers to focus on analysis of less structured data, where human analysis capability surpasses that of automated analysis.

Information sharing is also made simpler; digital data is easily disseminated over computer networks, with restrictions to access based only on availability of appropriate software and/or hardware, and certification for sensitive data. Public access to scientific data has significant impact on research [9, 121, 145]: data (in its original form, without pre-processing or filtering) is more easily obtained, allowing independent analysis to be performed by researchers using secondary sources of information, to confirm or disprove hypotheses formulated [18].

1.1 Data analysis

Scientific data is normally multi-variate, with a large amount of interaction between data elements [140]. Data from experimentation is often padded with additional information such as descriptions of experimental conditions. Data annotation, often employing different methods and sometimes conflicting terminology, further bloats the amount of data generated

in research, increasing resources required for its management. Data from new experiments may add to or contradict previous findings.

Challenges for data analysis include its management — efficient storage of raw and processed data and analysis results, in addition to the development of effective techniques for exploratory and more detailed analysis. This is impacted by methods used for recording and storing data; annotation of data varies depending on research fields, organisational standards and purpose for which research is being performed. Data exchange and retrieval is dependent on the different underlying schemas used for the large number of data stores available [19]. Development of tools employing intuitive methods for navigation through data is important for effective data analysis and IR [140]. Finally, data analysis requires effective methods for communicating its results [71, 105, 124, 165].

1.1.1 Visual data analysis

Limitations in the ability of humans to store and process especially large amounts of complex data (manually) mean that intuitive methods are required to aid data analysis and provide effective storage of intermediate and final analysis results [73, 130, 165]. This is important not only in scientific research: large amounts of complex, interacting data form a part of daily life [24], and require efficient management and analysis to retrieve knowledge important to both simple and critical decision making [95, 179].

Data mining, which employs machine learning for pattern recognition and feature extraction, is one option for uncovering information hidden within large data sets [72]. However complexity of analysis increases with data set size [2], reducing the ability to extract useful information.

Another solution that allows intuitive analysis is to visualise the information contained within data [49, 69, 71, 129, 163]. This does more than just present data and analysis results in visual form; visualisation reveals the inherent structure of data and highlights patterns and trends within it, harnessing the highly advanced perceptual ability of humans [39, 66, 129, 151], using vision, the primary channel for input, to reduce cognitive load in data analysis. [179] talks about the power in “*seeing information*”; graphical representation of data is often more compelling and memorable than the textual equivalent [86], and provides a powerful method for communicating data structure and content [83], and results of the analysis to an audience.

Information visualisation provides overviews of what is often very large amounts of complex, abstract data, mapping its semantic content to a spatial representation to allow intuitive encoding of data attributes [39, 163], employing simple physical properties such as colour, as illustrated in [74], shade, hue, saturation, density of data elements [49], and shape and size of physical objects [110]. Relationships within data are more easily recognised, leading to identification and further analysis of regions of interest (ROIs) within the data.

1.1.2 Bioinformatics and biological data analysis

Digitally generated data often requires different methods for analysis than those used for data obtained from traditional experiments performed at the laboratory bench. A good understanding of the data being analysed and how results of analysis are to be used are necessary in determining which tools or techniques are best suited for analysis, or to extend existing or develop new tools to provide optimal analysis [110, 144, 158]. *Bioinformatics* was born out of the need to provide specialised computational methods for digital biological data analysis.

Bioinformatics involves multiple disciplines, comprising mainly biologists and computer scientists, working together to develop optimal methods for extracting knowledge stored within biological data. Different researchers with varying backgrounds bring multiple perspectives to data analysis, revealing more insight into data than a restricted set of research fields would [105, 121].

1.1.3 Ontologies in data analysis

The *Free On-line Dictionary of Computing*¹ (FOLDOC) defines an ontology as:

1. *An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them.*
2. *The hierarchical structuring of knowledge about things by subcategorising them according to their essential (or at least relevant and/or cognitive) qualities.*

The multiple disciplines involved in bioinformatics increase the probability that different terms will be used to describe the same concept, or the same or similar terms for different concepts. Ontologies store semantic information in a knowledge domain, aiding data exchange and knowledge transfer by providing a reference framework that promotes consistency in data interpretation [19, 29, 67, 90, 93, 171]. Ontologies describe a knowledge domain and the relationships that occur between and within elements belonging to this domain [9]. [14] go on to define the role that ontologies play in data analysis — using the knowledge ontologies contain to annotate data, aiding analysis by associating semantic content with data, and easing comparison of different data sets and by different disciplines, benefiting both manual and automated data analysis and IR.

This thesis looks at how research in information visualisation can be used to aid data analysis in genome research, harnessing skills and knowledge that the different research areas involved contribute to obtaining new knowledge in the field. Findings are applied to the visual analysis of ontology data, to aid data interpretation and integration, illustrated using sample data from the EMAP and XSPAN projects (refer § 1.2 and 1.3).

¹FOLDOC can be accessed at: <http://wombat.doc.ic.ac.uk/foldoc>

1.2 The Edinburgh Mouse Atlas Project

The Edinburgh Mouse Atlas Project, EMAP, stores information on the developmental stages of the mouse using hierarchically structured, text indices mapped to reconstructed 3D models of mouse embryos. This provides a spatio-temporal framework that tracks normal development of the embryos [13], for research into the genetic makeup of the mouse.

1.3 The Cross-Species Anatomy Network

Determination of the structure and function of newly discovered genes is aided by the comparison of gene expression data for corresponding components in different species [10, 137]; evolutionary conservation results in similarities in genes derived from equivalent components in related organisms [9, 20]. The Cross-Species Anatomy Network, XSPAN, is developing a system for integrating multiple data sets, to aid the determination of relationships across different organisms starting with the anatomy ontologies for the model organisms mouse, human, *Drosophila*, Zebrafish and *C. elegans* [32, 31], based on similarity in genes expressed in cells, tissues and organs.

A detailed look at the EMAP and XSPAN projects can be found in chapter 5.

1.4 Structure of thesis

1.4.1 Identifying issues pertinent to visual data analysis

One of the main challenges in data analysis is obtaining overviews of data that aid users in building effective mental models of data structure, an important component in understanding and retrieval of the information contained within data and the relationships that occur between data elements. What constitutes an effective overview however varies, depending primarily on user information requirements — useful presentation of information that answers the questions being asked by the user can only be obtained if these requirements are effectively presented and correctly interpreted. User requirements can then be translated first into more general tasks such as the need for exploratory navigation through data to obtain a general overview of data structure, and then broken down further into, say, requirements for more directed search and query, with the ultimate aim being to retrieve specific information from data.

Domain knowledge, user backgrounds and skills brought to analysis, and data type and amount influence what types of visualisations will be most effective for analysis. It is difficult to obtain a generic solution that will cater to all needs, and the designers of an analysis tool or even its users will not always be able to determine the ideal amount of information to feed into data overviews. It is difficult also to achieve a good balance between providing enough information to allow users to obtain an overall appreciation of data structure, and too much, resulting in visualisations that are cluttered or that lead to cognitive overload in

users. Interactive abstraction that allows users to filter out data of lower relevance provides a powerful method for customising visualisations to suit varying user needs. This may then be extended to detailed analysis of ROIs, with the introduction of perceptual and other cues that map to user needs and ability.

Chapter 2 reviews research in information processing in humans and the bearing this has on performance of data analysis. This leads to a discussion on harnessing perception in humans to obtain intuitive analysis, and an exploration of the different techniques available for visual data analysis. The merits and limitations of a sample of data analysis tools are examined, assessing also alternative solutions developed to overcome the limitations identified. A comparison between two and three-dimensional (2D and 3D) visual data analysis is then presented, and the chapter concludes with a brief discussion on existing support for remote analysis using dedicated networks and the Internet. Chapter 3 continues to look at graph visualisation and its contribution to visual data analysis. A sample of graph visualisation tools is reviewed, with a focus on hierarchical (data) visualisation. The chapter ends with a discussion on the limitations of tree graph visualisation.

A recurring theme is the ability of most tools to satisfy only a small sub-set of the large number of information requirements identified in different fields, for different data types and formats and for varying end uses. A choice has to be made between developing tools that provide simple overviews at the expense of limiting detailed analysis, and applications built to satisfy specific needs but that are difficult to extend outside a very small focus. Challenges in effective capture of user requirements, especially where they span a large number of target groups and disciplines, mean that most tools are likely to fall in the latter group, catering to a small set of users. A danger here is that such tools may not adhere to common standards for data exchange, further limiting reuse and the potential for combination of individual tools to create workbenches with wider applicability.

1.4.2 Involving the user in developing a solution

Attempting to provide a general, overall solution for data analysis is not a practical option; starting from a smaller sub-set of user information requirements and designing scalable, modular tools should result in more (re)usable and potentially extensible solutions. This project restricts initial analysis to independently created but inter-related data sets in a biological domain. Challenges in analysis of this data are those typically encountered in research and data analysis: data sets are often created independently, may be stored in databases with different or even incompatible underlying schemas, data may vary in format and accuracy, and employ different terminology for defining and annotating data elements and relationships between them. That the data being studied is stored using ontologies shows that an attempt is being made to reduce differences in data storage and presentation; however even ontologies in a specific domain still contain marked differences in terminology. Chapter 4 discusses the contribution ontologies make to research in bioinformatics, looking

at how ontologies are used to improve data exchange and analysis. Tools developed for analysis of bioinformatics data are reviewed, followed by a look at applications of bioinformatics. Chapter 4 concludes with a brief summary of the ethical issues in bioinformatics and especially genetics research.

Chapter 5 describes the data analysis issues this thesis seeks solutions to, performing research in bioinformatics. General requirements for analysis include the need to provide overviews that aid the determination of overall data structure, and further detailed analysis, to:

1. identify (implicit and explicit) relationships in data and the structures these define
2. determine equivalence between elements across multiple data sets
3. trace continuous or temporal relationships in data.

The two projects EMAP and XSPAN, which make use of anatomy ontologies, provide a practical domain in which to test the ideas explored, allowing an existing data set and typical target users to be used to evaluate the options proposed for analysis. Challenges for analysis in the two projects are similar to those typically encountered in the use of ontologies in general, and in biological research that makes use of anatomy ontologies as a common base from which to compare data in related fields. Solutions found to these requirements may be extended to similar research in bioinformatics, and the results of analysis to wider research in biology and other related fields.

The remainder of the thesis details the process followed to address the analysis requirements identified. Involving users in the determination of usable solutions for visual analysis required a physical prototype that could be used to articulate design requirements and also help to overcome the *language* barrier between the different disciplines involved. Chapter 6 details the design and development of an initial prototype using simple node-link, directed acyclic graphs (DAGs) to provide overviews of the data sets under study. Extensions to existing techniques that provide solutions to the problems inherent in the use of hierarchical graph visualisations are described, followed by a heuristic evaluation carried out to determine usability of the initial prototype. Using data from the EMAP project to test different options for analysis a number of issues in visual analysis were examined. General difficulty capturing user requirements has been previously mentioned; this is complicated further in this case because of the different disciplines involved in bioinformatics. Software development, dominated by computer science (CS), involves design and development of complete tools or modules; graphic design may play a role in the creation of visual aspects of interfaces; both biologists and computer scientists would be involved, to varying degrees and for different aspects of the actual process of data analysis, dependent on the skills each possesses. Collaborative, continuous analysis is the norm; ensuring users are able to pick up from previous work or that of others requires ability to place markers in data, providing annotation that points to information of interest or changes made to underlying data, for instance. Catering to the individual needs of different users also involves the determination

of effective cues for visual querying and IR. Working with multiple data sets further requires transparent mappings between the different sources of information, to hide differences in terminology and underlying structures of the data, and aid users in identifying keywords and formulating queries that are able to retrieve information contained in data.

Chapter 7 details the preparation for and process used to carry out a structured usability evaluation of the visualisation prototypes developed, based on user information requirements and information obtained from the heuristic evaluations previously performed. A discussion of the results of the evaluation leads to suggestions for additional functionality for improved data analysis. Chapter 8 describes changes to the prototypes, building on the techniques developed to aid the identification and analysis of relationships in the anatomy ontologies.

1.4.3 Research findings

Chapter 9 describes a final evaluation of the applications built, reviewing first the issues for analysis previously identified. It then goes on to examine further questions brought up during the first set of user evaluations:

1. marked differences were found between navigation and search strategies based on user backgrounds and domain knowledge. The need to identify cues and other analysis aids that each target user group would make optimal use of was critical to the usability of the final solution developed.
2. spatial awareness plays an important role in the use of data visualisations. How could visualisations be built to harness different levels of spatial ability to extend cognition and help users form effective mental models of data structure and an understanding of relationships within the data?
3. arguments exist for restricting dimensionality of visualisations to two, to make optimum use of the 2D displays available in the average working environment. However despite inherent complexity in 3D it may provide more intuitive analysis than 2D beyond simple requirements for space in which to view large, high-dimensional data sets. How could visualisations be built to optimise both use in 2D and extensions to higher dimensions?

This evaluation also measured the extent to which specific requirements for analysis (detailed in chapter 5) had been met. Having confirmed in the first structured evaluation that 3D provides advantages for analysis of the data beyond the capabilities of 2D the second structured evaluation went on to test performance of more complex tasks in 3D, to determine additional aids required for what is recognised to be more complex navigation and exploration. Additional research also looked at the influence of spatial awareness on the ability to navigate effectively through 3D space and perform intuitive data analysis and IR.

The thesis concludes with chapter 10, which reviews the application of information visualisation to the analysis of anatomy ontologies, to provide intuitive data exploration and

improved ability to identify relationships within data. This thesis has found that novel approaches to visualising information are necessary, to harness differences in users and data especially in cross-disciplinary fields such as bioinformatics, if effective analysis is to be obtained. Working with users to discover how best to address their information requirements it was found that a fine and constantly changing line is followed trying to identify how much and which information contained within data should be extracted and fed into the generation of visual data structures, and how supplementary text should be woven into information spaces to aid interpretation of visual representations of data. Empowering users with intuitive, customisable and extensible tools provides the base required to perform effective analysis, allowing semantically meaningful representations of the information content of textual data to be built using physical objects that map to navigation and exploration in the real world users are familiar with.

Finally, the discussion in § 9.6.1 on the limitations of the approach used leads to suggestions for further work to provide improved support for visual analysis.

Chapter 2

Data analysis and information processing in humans

2.1 Information processing in humans

Limitations in humans for information processing mean that cognitive overload quickly occurs in analysis of large amounts of complex data [165]. Solutions that make use of graphical representations of especially complex data have been proven to aid analysis and problem solving, augmenting memory and transferring cognitive effort required to more intuitive perception [6, 66].

2.2 Data analysis

No one tool or technique is ideal for all aspects of data analysis; different tools work best for different data sets, dependent on information required, the purpose for which information obtained is to be used, the current stage of analysis, data types used to store information and domain knowledge and analysis skills of users [108, 117, 143, 144]. Which tools are used for data analysis influence strongly what information is retrieved and how it is interpreted [144]; different tools and analysis methods will highlight alternative perspectives of data, uncovering different aspects of information stored [78, 114, 171]. It is often necessary to use multiple tools and/or techniques in concert [108], sometimes iteratively [49], to harness the features of each to obtain optimal analysis and retrieve knowledge stored in data. Using techniques or tools inappropriate for the analysis required may lead to misleading conclusions, and result in non-valid hypotheses and theories. It is important to note that it is not necessarily the information retrieved that results in incorrect decision-making — interpretation of data in one knowledge domain or for a specific purpose may not necessarily apply in other situations [171].

Ontologies provide support for the cognitive effort required for effective data analysis and interpretation, by serving as semantic frameworks applicable to different knowledge

domains. Following on from prior research and anecdotal evidence that illustrate the highly advanced perceptual ability of humans, this thesis looks at using visualisation of ontologies to aid understanding of interaction within complex, related, multi-dimensional data sets. This chapter reviews data analysis in general and visual analysis in particular, and chapter 3 looks at graph visualisation. Chapter 4 concludes the literature review with a discussion of techniques available for bioinformatics data analysis employing ontologies.

2.2.1 The process of data analysis

Data analysis can be broken down into four main parts:

1. data collection and storage
2. data pre-processing
3. exploratory and detailed analysis
4. presentation of analysis results.

Data collection and storage

How information is collected and stored has a significant effect on tools that can be used to analyse it; differences in data format, accuracy and quality may restrict analysis to specific techniques and tools, and will also determine how easily information stored is retrieved. Annotating data using ontologies or controlled vocabularies aids the process of IR [9]; it removes ambiguity inherent in heterogeneous data, and aids automated pre-processing and analysis, and dissemination of data.

Data pre-processing

This includes data reduction, filtering out noise, or suppressing superfluous or non-relevant data [2, 122, 124] to reduce complexity and prevent occlusion of useful information, and reveal patterns and relationships within data [49, 74, 163]. Different methods for storing data often result in variations in formatting, accuracy and annotation. Before performing analysis it is often necessary to convert data to formats suitable for analysis tools being used [122]. Meaningful comparison of (multiple) data sets may also require normalisation of data [124].

Exploratory and detailed analysis

Exploratory analysis is often used to obtain a broad idea of information contained within a data set and that may be extracted, to determine which methods and techniques are most appropriate for more detailed analysis. Though not dependent on preconceptions about data content or structure it is still useful to bring domain knowledge to bear in exploratory data analysis [8], especially where specific knowledge or validation for hypotheses is sought. Data mining employing visualisation is especially useful for exploratory analysis of large data sets [71].

Detailed analysis looks to retrieve information stored in data, highlighting relationships that occur within data and flagging anomalies identified. Automated data analysis, which is more efficient than humans for performing tedious, repetitive computation [2], may be augmented with options for interactivity that harness perception in humans for analysis and identification of patterns not recognised by automated algorithms.

Multiple tools and techniques are often used in concert [124]; different storage methods and formats, user data analysis skills and information requirements all play a part in determining which techniques will provide intuitive analysis and IR. Scalability of tools, support for interactive or automated, intelligent analysis, metaphors used in design, and customisability are all important factors in choice of tools used for analysis [66]. Methods used to analyse large data sets include data summarisation and reduction, or flattening. Visualisation allows data to be laid out such that relationships within it are highlighted, using among others, scatter plots and metaphors from real life that map complex data to trees and information landscapes. Techniques commonly employed in complex data analysis are discussed in § 2.3 and § 2.5.2, with a focus on visual analysis.

Presentation of analysis results

Written reports and summaries are traditionally used to present results of data analysis. Visualisation provides a more intuitive method for presenting conclusions drawn from analysis, using alternative options to allow interpretation suitable to different target audiences [124].

2.2.2 Collaborative work and incremental data analysis

Design of data analysis tools must take into account collaborative aspects of analysis, to enable and foster user interaction [39]. Data analysis is very rarely performed in isolation; a data set may be analysed simultaneously by multiple researchers, or individual users may be one in a sometimes multiply-linked chain working on a different aspect of analysis of a single data set.

Incremental, exploratory visualisation is useful in two main instances: where data is constantly updated, and to make use of results of previous analysis. For the former, incremental analysis allows temporal characteristics of data to be visualised, highlighting changes in data structure with time. As new properties or characteristics of data are received classification of objects may change, revealing new relationships and presenting alternative perspectives of data.

Continuous or even independent analysis of the same data set by multiple researchers benefits from previous analysis performed [66], preventing unnecessary repetition of identical processes and leading to savings in time and financial cost [122]. Markers may be placed in data and analysis results to flag information of interest or to point to other data repositories, annotation may provide additional information based on prior (domain) knowl-

edge of different analysts, resulting in improved navigation and IR [39]. Support required for collaborative work includes management of data conflicts due to simultaneous access to data. Incremental analysis further requires support for data updates in real time.

2.3 Visual data analysis

Generating effective, intuitive visualisations for high-dimensional data remains a challenge for data analysis: obtaining a useful overview of the large amounts of what is often complex, abstract data, and providing options for navigation that prevent disorientation and aid exploratory analysis. Analysis therefore often focuses on only local data immediately relevant to users' information requirements, while relationships with other more distant data may remain undiscovered [140, 152].

2.3.1 Information visualisation

An understanding of how humans process information is important in the creation of visual representations of data, if they are to ease analysis [81, 151]. Information visualisation brings together learning from multiple disciplines, including cognitive psychology, human-computer interaction, computer graphics and art [129], to harness human perceptual ability for effective, intuitive analysis. Visual data analysis systems are supported by database management and multimedia systems, networking for remote analysis, and often make use of interactivity and animation [108].

Improvements in technology have led to lower costs for larger amounts of computing power, disk storage space, high resolution displays, advanced graphics cards and more sophisticated software for data visualisation and analysis. Tools incorporating novel techniques are constantly being developed for different aspects of visual, computational data analysis [163, 184]. More effective analysis is possible for increasingly large amounts of complex, heterogeneous data, using tools that extend intuition and perception in humans.

Data is often stored as text. Though this may simplify transfer and exchange there is a high cognitive load associated with analysis of large amounts of especially complex data in textual format. It is difficult to obtain an overview of data structure, and relationships within the data are not easily recognised. Information visualisation maps abstract data to a spatial representation, to aid the formation of effective mental models of overall data structure [71, 96, 129]. Users are able to *see* the structure of data, whereas text is only able to *describe* data; mental models formed from text may be neither accurate nor complete [6]. Visualisation transfers the conscious cognitive effort required for complex data analysis to the more intuitive perceptual system [108], which is able to recognise patterns and trends in data more easily [78, 83, 151], leading to increased understanding of information contained within data [83].

[163] stresses the importance of his “*Visual Information-Seeking Mantra*” for effective data analysis and IR:

“Overview first, zoom and filter, then details-on-demand.”

Visual (semantic) overviews of data should be generated that provide context to support navigation and exploratory data analysis [66, 110], followed by more detailed analysis of ROIs [160]. (Interactive) visualisation offers the opportunity to look at multiple aspects of a data set in parallel, from different points of view.

However data overviews are only useful if they are able to capture the structure of an entire data set, which is easily achieved for relatively small amounts of low-dimensional data. For large, complex, multi-variate data sets, abstraction, dimensionality reduction and/or summarisation may be necessary to obtain usable overviews that can be displayed on the 2D surfaces commonly used for visual data analysis [117].

Further detail should be made available as users approach ROIs to perform more detailed analysis [39, 117, 163], preferably within the context of the overview. Interactive visual analysis allows perception to be used to build an understanding of data structure during (exploratory) navigation, aiding the formation of mental models of overall data structure [165]. Combined with intuitive encoding that employs simple physical properties such as colour, saturation and shading, patterns and trends within data are revealed that may not be recognised as easily in text.

2.3.2 Metaphors in visualisation

Visualisations that map directly to real life are useful for (scientific) visualisation of spatial data. However this is not necessarily the case for the abstract data that is used in information visualisation, as it does not map naturally to a spatial representation. Visual metaphors effective for analysis that users recognise and understand can be used to encode non-spatial data to provide intuitive analysis [63, 129]. Here too, the data types being analysed, the analysis to be performed, and the purpose for which analysis is required are important in determining metaphors most appropriate for visual analysis. User skill and domain knowledge also play a role in the choice of metaphor(s) used [24, 117].

An added advantage in the use of visual metaphors is the ability to extend them or apply magic effects to provide options for data analysis that would not be possible for the equivalent objects in the real world [91]. Generating visualisations that provide intuitive interaction with data and that lead to effective IR is more important than remaining faithful to metaphors on which visualisations are based [66, 164]. Physical objects representing data in visualisations may be *rubber-banded*, for instance, allowing more flexible manipulation of data than would normally be possible [48]. Spatial metaphors may be extended to allow users to *fly over* and *through objects*, and *magic lenses* may be used to reveal semantic detail in data content.

The following sections describe metaphors commonly used to aid interaction with complex computer systems and provide options for intuitive data analysis. See also [57], who provide a compendium of data visualisations that illustrate some of the metaphors described

here.

Spatial metaphors

A well-known example of a spatial mapping is the *rooms metaphor*, which maps the semantic content of data to rooms in a building, to aid querying and IR. [39] illustrates this using an architectural metaphor, relating the design of physical spaces to the varying needs of individuals. Differences in type of work and variations in tasks performed as part of normal work require custom design to ensure rooms are built that suit different needs and methods for working, even within the same building (or context). This metaphor takes interaction between different users into account, which exists despite differences in working practices; individuals move between physical locations; any one room or building will only form a portion of their (working and social) lives.

Geographical metaphors

The landscape metaphor is illustrated in [39], using peaks, valleys and shorelines in a natural landscape to encode data. Areas of high relevance have a higher density of data nodes, peaks and valleys represent rough ground, where data may be less reliable than that found in smooth areas. Shorelines are used to represent less important data, and islands contain outliers. Fog may be used to fade out distant data of lower immediate relevance. Natural landmarks such as rivers and other user-defined boundaries such as borders may be placed within visualisations to prevent or reduce disorientation during navigation and to serve as markers for continuous analysis. [44] stress the influence of landmarks on intuitiveness of navigation, and the ability to form a good understanding of data structure.

Interactive visualisations may extend the landscape metaphor to allow users to *fly* over data to obtain an overview of its structure, descending to the level of the visualisation to immerse themselves in and examine ROIs in detail.

The cityscape metaphor uses a 3D visualisation to lay out hierarchically structured or network data. [116] illustrate the metaphor using rectangular blocks similar to buildings to represent data elements as shown in figure 2.1. The visualisation is laid out on a 2D plane in 3D space, so that an overview of data structure is easily obtained from any angle. Properties of data nodes are encoded using size, colour and position of blocks drawn.

Users are able to map navigation through urban areas to exploration of the information space this metaphor creates, by reading maps, using landmarks such as familiar buildings and road signs and recognition of boundaries between locations.

Organic metaphors

Organic metaphors, which are especially useful for visualising hierarchical and temporal data, map growth and withering or dying away of parts of an organism to changes in data

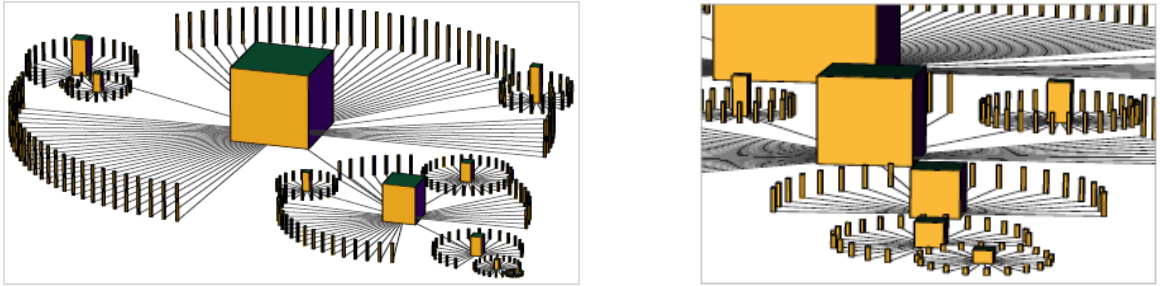


Figure 2.1. An overview of a cityscape is shown on the left-hand side, while the image on the right zooms in to examine detail in an ROI. (Images courtesy of [116])

with time. [77] describes the use of an *anemone* to visualise continuous change in interest in data while browsing through a web site. Growth in length and diameter of tentacles is proportional to observed interest in data. Figure 2.2 shows a series of snapshots of an applet that uses the *anemone* metaphor to visualise dynamic update of use of a website.



Figure 2.2. [77] uses “*organic information design*” to visualise dynamic update of the use of a web site, by mapping growth of a biological system, the *anemone*, to a constantly changing data set. The brown lines show paths already traced, while the tentacles of the *anemone*, the pages in the site, are shown in white, with usage mapped to size of each tentacle. (Snapshots printed with permission from a run of the *anemone* applet at: <http://acg.media.mit.edu/people/fry/anemone/applet> (last viewed Jul 2006).)

The gardening metaphor extends growing in plants, pruning and weeding to especially hierarchical visualisation, to deal with the occlusion common to large data sets. Branches may be pruned to suppress data of lower importance and improve navigation [127, 152], and then re-grown to reveal hidden data. Flowers may bloom or fan out, and weeding may be likened to filtering out non-relevant or superfluous data.

Trees provide an abstraction commonly used to represent hierarchical data in visual format. Trees are especially useful for progressively clustering data into composite nodes, revealing more detail as users approach ROIs. This has a two-fold purpose: to aid navigation by reducing the disorientation that occurs during exploration of dense information spaces, and to provide a natural method for data classification, based on user-specified or automatically derived criteria.

Physical metaphors

Fish-eye views occur naturally in all aspects of life: local ROIs are given greater significance than the equivalent in more distant areas. An example can be seen in the relationships occurring within an organisation: interaction between employees decreases as one moves from the members of a team to the larger department and the organisation as a whole, depending also on relevance of others to a specific employee's work. [80]. [82] also relates the fish-eye concept to IR; semantically related items are more likely to be identified when users are prompted with terms or concepts describing information of interest, even where such items have large physical separation.

Fish-eye projections make use of a wide-angle lens to increase magnification as one approaches the focus of the lens [159]. This provides what is known as a *Focus+Context* (F+C) technique for detailed analysis of ROIs within the context of the overview.

Magic lenses which are based on the metaphor of a magnifying glass, transform the area of a visualisation over which they are placed [170]. The metaphor is extended to allow size and shape of lenses to be modified interactively to suit different requirements. Multiple lenses can be combined to perform complex physical or semantic transformations on data. Compound filters or queries can be created without the need to learn query syntax, a benefit for users with limited programming ability or experience in complex querying [140]. Lenses can also be used simultaneously in different areas of a visualisation to provide multiple foci, allowing the comparison of ROIs to reveal relationships that exist between them. Alternatively, multiple windows can be used to compare different perspectives of the same ROI, analysing different attributes of data or changes to data with time, each generated using variations in lens combinations.

Transformations to data may be temporarily applied, with ROIs reverting to their original state when lenses are removed, useful in exploratory data analysis. Alternatively transformations may be saved to an alternate view [170], and retrieved for use in other analysis sessions. Modifications to data may be applied permanently, resulting in changes to the underlying data and/or the visualisations generated. An advantage in magic lenses is that transformations are locally applied, to data in ROIs, preserving the structure of the overview. Lower requirements for resources for computing transformations have a positive effect on system response. A marked disadvantage, however, is that data immediately surrounding the focus and ROI is obscured by the lens.

Spring layouts are based on a system of springs between data elements, with forces of attraction or repulsion based on (dis)similarity between data [89]. From a random layout of nodes containing a large amount of energy the springs settle down at equilibrium to produce a scatter plot that maps physical distance between nodes to semantic similarity. Clustering of like data naturally occurs, based on criteria used to set forces between springs, illustrated

in [74], who use a spring-based algorithm to lay out data in their *Cluster Map*.

The main advantage in spring layouts is that even if the process of layout optimisation is interrupted before it comes to completion, an approximate layout is still obtained.

Composite Metaphors

In addition to stretching individual metaphors it is sometimes useful to combine multiple metaphors so they complement each other, building richer representations of data than individual metaphors are able to [24, 91]. A fish-eye view may be mapped onto a geographical metaphor, to obtain the appearance of a raised surface on a 2D plane as for a globe stretched out in 2D. [117], for example, map a hierarchical graph to a hemisphere with a moveable focus, to obtain an F+C system that also harnesses the additional degrees of freedom for navigation in 3D. [24] describe how a *magic* effect similar to the use of portals may be combined with a cityscape to simulate use of an underground transportation system. They also describe a *night sky* metaphor with portals leading to different points in space. In contrast to the busy cityscape, this is a sparsely populated information space containing widely separated objects, so that the ability to *jump* between distant locations is even more important for intuitive navigation between data objects.

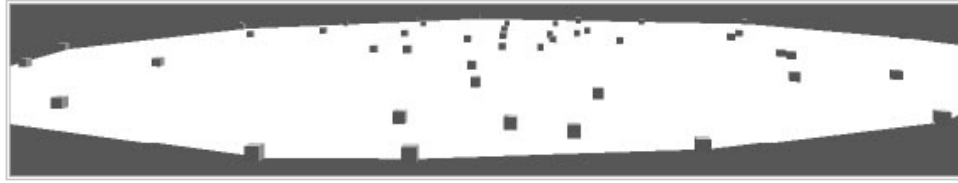
2.3.3 Navigation through data

Using visualisation to reveal the structure of especially complex data helps to combat the disorientation that is often encountered navigating through large data sets. This allows exploration of the information spaces obtained, during which process perception and intuition are used to build an understanding of the structure of the visualisations, to retrieve knowledge stored in the underlying data. Mapping paths through data helps to highlight interaction between data elements, creating bookmarks, landmarks and/or history sessions that serve as an aid in incremental or continuous analysis and provide quick access to previously discovered information [125, 24]. Markers also help to reduce disorientation by providing recovery or orientation points in data, especially useful in exploratory analysis. Multiple cameras and viewpoints, which may also serve as landmarks, also provide different perspectives of information.

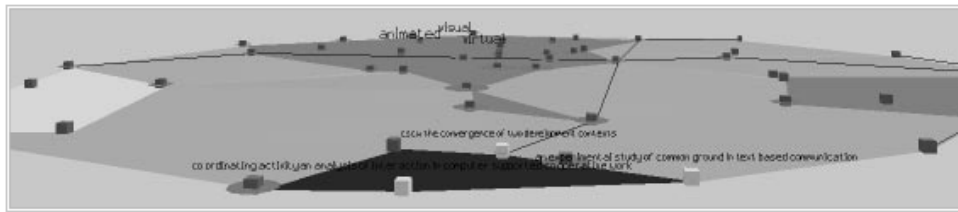
Clustering related data into composite nodes, and revealing more information as one approaches ROIs, helps to focus on data of interest during exploration [127, 159]. Fading away increasingly distant (and less important) data removes the distraction it poses, also reducing the occlusion that leads to an increase in disorientation during navigation. Care should be taken to ensure that visual cues are used that provide effective descriptions of semantic detail hidden within data, so that users are able to retrieve information sought [24].

[40] provide a good illustration of the effect of visual cues on navigation and exploration through a data set: figure 2.3(a) shows a scatter plot laid out on a flat surface. The only

cues available are depth with distance obtained using natural perspective. Figure 2.3(b) adds shading to the scene, borders and supplementary text describing objects of interest, to form an information landscape that creates a natural map through the data.



(a) A landscape that provides only depth and distance cues to aid data interpretation



(b) Borders, shading and supplementary textual detail provide a rich information landscape

Figure 2.3. Very little information can be obtained from the visualisation in figure 2.3(a) because of the lack of visual cues or supporting text. It is however transformed by the addition of simple cues: shading in greyscale, borders and supplementary text, in figure 2.3(b). (Images reprinted with permission from [40])

2.3.4 Issues in visual data analysis

Poorly constructed visualisations may mislead rather than aid interpretation of data [49, 56]; although graphic design plays an important role in the generation of effective visualisations a visually appealing image is not a substitute for effective encoding of information stored in data [108]. An additional complication is ensuring that visualisations generated are able to meet the different information and analysis needs of users who may have marked differences in domain knowledge and data analysis skills [24].

A classic example of poor visualisation resulting in incorrect analysis is the plot of O-ring failures with temperature that led to the crash of the space shuttle Challenger¹ (last viewed Jul 2006). The initial plot of O-rings with temperature excluded important contextual data, so that conclusions drawn were erroneous; plotting the complete data set revealed a pattern quite different from that in the original plot.

Overview versus detail

Visual overviews are an important aid in the recognition of the overall structure of a data set. However, one of the most significant problems information visualisation suffers from is poor scalability, often displaying an exponential increase in occlusion with data set size.

¹See the Report of the Presidential Commission on the Space Shuttle Challenger Accident at: <http://history.nasa.gov/rogersrep/genindex.htm> (last viewed Jul 2006). See also Michael Friendly's *Gallery of Data Visualization* at: <http://www.math.yorku.ca/SCS/Gallery>

This may render overviews of especially large data sets unusable and increases difficulty in data analysis [163].

Methods commonly used to reduce occlusion include clustering of like data into composite nodes, based on user-specified or automatically determined criteria for similarity, to reduce the number of objects drawn to the display. Another solution is to magnify visualisations generated to force elements apart, reducing occlusion due to overlap of data nodes and labels. This is an advantage for analysis of ROIs in isolation, since more detail can be seen. However because display size remains constant a large portion of the resultant visualisation may run off the screen. It becomes necessary to translate the viewpoint to move between ROIs, using scrolling and panning in 2D, and additionally, rotation in 3D. The context of the overview is lost [80, 125], increasing difficulty in navigation and the likelihood of disorientation occurring.

One way to regain context is to use a coupled window that retains the overview, while studying ROIs in a separate window [69, 160]. However for small screens or displays it is difficult to work with more than one window at a time without a large degree of overlap. Further, there is additional cognitive effort required to map between the visualisations in separate windows [82].

F+C techniques such as fish-eye views provide an alternative solution [98]: the ROI is magnified and the overall visualisation redrawn in the same amount of space, with successively lower levels of magnification as one moves away from the focus. Such techniques however distort the layout [82, 165], and may destroy users' mental models of data structure, leading to an increase in cognitive load during analysis. Using a magic (Cartesian) lens provides magnification to ROIs without distorting the original visualisation; however the area immediately surrounding the ROI is obscured [125, 165].

Analysis of regions of interest

Properties of and values of properties for individual data elements may vary widely especially in large data sets. Large (spatial) distances between data nodes due to the criteria used for layout also make it difficult to perform comparisons based on physical properties such as size. [48] (see figure 2.4) employ interactive control over the visualisations they generate to modify scaling in user-specified sub-sets of data, encoding differences in scale using colour.

Alternatively data outside ROIs may be suppressed using different techniques, with user choice dependent on skill, information requirements and data structure, among others. Applying different levels of transparency to data as described in [138], or progressive fading out of data is commonly used to remove the distraction of data outside ROIs. Colour, hue and saturation may also be used to encode data attributes, as illustrated in [98] and figure 2.5, providing visual cues that aid data analysis.

Extraction of ROIs for analysis in isolation allows more distant data to be brought closer to the user's viewpoint (as illustrated in figure 2.6). This serves two functions: it removes the

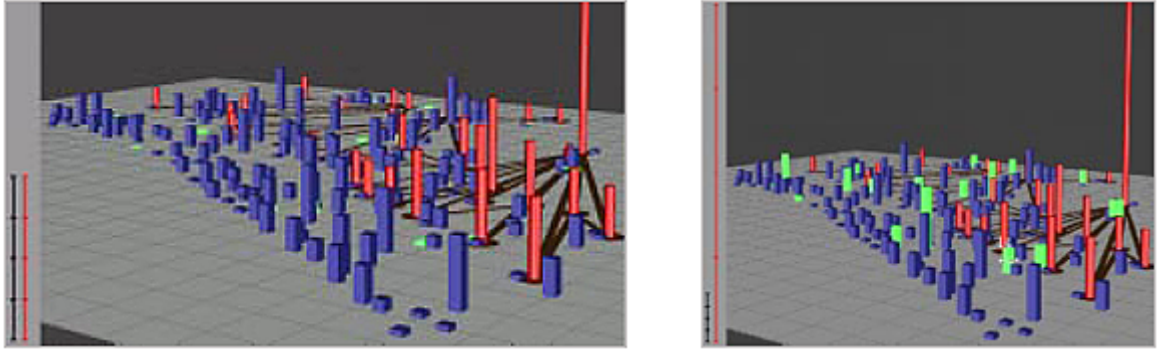


Figure 2.4. The visualisation system developed by [48] uses non-uniform scaling to highlight subsets of interest, within the context of the overview. The diagram on the right rescales the heights of the data elements for the sub-set highlighted in green; compare with the data set with uniform scaling for all elements on the left.
(Image courtesy of [48])

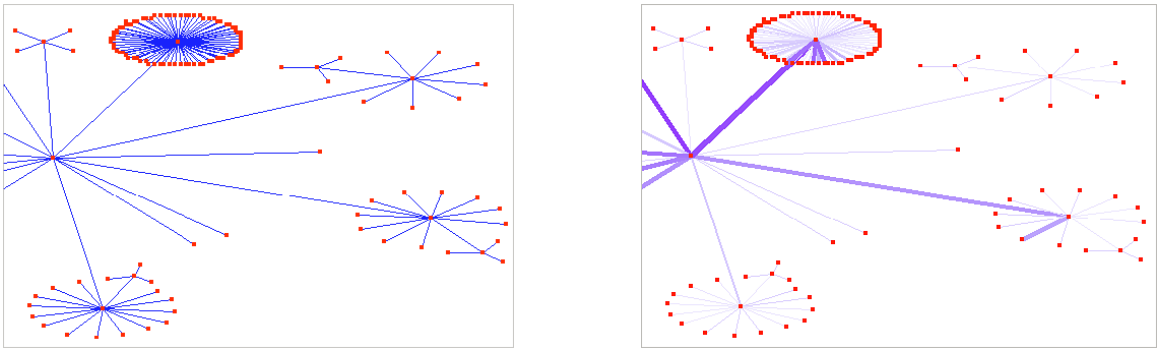


Figure 2.5. Variation in colour and thickness of links between nodes is used to highlight paths leading to ROIs in the graph on the right [98]. Without these additional cues in the graph on the left it is difficult to determine where data of specific interest may lie.
(Images reprinted with permission from [98])

distraction of surrounding data of lower interest, and brings widely separated data elements closer together. To maintain context [48] use *skeletons* to mark the original positions of these data nodes. Other options include suppressing non-relevant data (see figure 2.7)) and hiding it altogether. The latter however isolates ROIs, while the former allows some degree of context to be maintained, while still minimising distraction of less important data and significantly reducing occlusion.

Maintaining mental models of data structure

Reproducibility is an important requirement for visual analysis [49]. Graph layout should remain consistent for multiple runs of the same algorithm if consistent mental models of data structure are to be formed that aid understanding of data. Predictability of the structure of visualisations generated is necessary for analysis to be repeated and theories validated [99, 127]. Force-based graphs such as spring layouts (discussed in § 2.3.2) pose a problem, as different results may be produced for each run of an algorithm, especially for cases where the solution does not reach the global optimum [89].

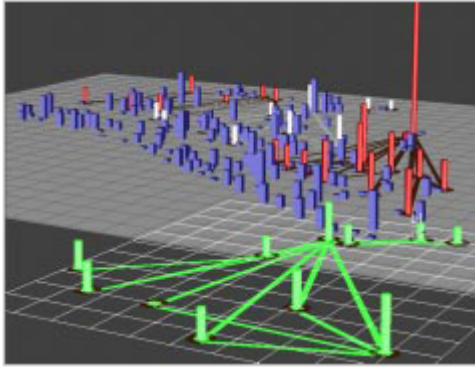


Figure 2.6. Extracting an ROI from the overview for detailed analysis in isolation. (Figures 2.6 and 2.7 courtesy of [48])

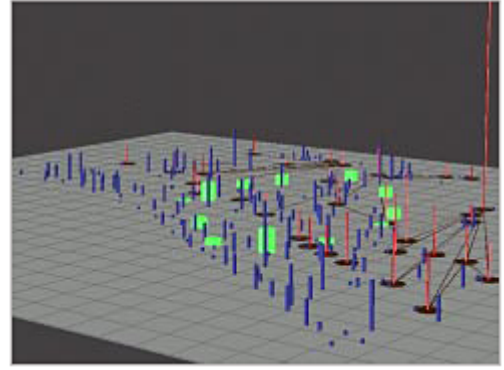


Figure 2.7. Suppressing data surrounding the sub-set selected by scaling down the width of data elements. This simultaneously highlights the sub-set of interest.

Managing complexity in visualisation

Although images can store a large amount of information there are limits beyond which complexity of visualisations may render them difficult to interpret. It is also often necessary to supplement visualisations with text labels or annotation [49, 86]. This helps data interpretation and analysis, and confirms conclusions drawn from images, clarifying areas that may not have intuitive interpretation.

Employing minimalism also helps to lower complexity, easing analysis in addition to reducing resources required to generate visualisations [78, 97], making it easier also to identify important points in data and those elements that do not follow a general pattern.

2.4 Browsing, searching and querying data

Research can involve extracting useful, previously unknown information from large data sets. One method for achieving this is to browse through the data, to obtain some idea of the information stored in it. A more direct approach is to query data for specific information, especially useful for very large data sets, where disorientation and cognitive overload commonly occur during navigation through the data [38].

A major limitation in querying, however, is learning the complex syntax required to filter out non-relevant data and retrieve information required [3, 165]. Further, querying data from multiple sources is beleaguered by a lack of integration in underlying schemas of data stores [19, 84]. This often means the need to reformulate queries to suit different data structures, in addition to determining appropriate search terms that suit terminology used in data stored. Developing systems that can parse natural language reduces the need for pure query syntax, and provides an IR aid for especially non-technical users.

Further, querying involves a knowledge of the structure of data and a fair idea of what information might be stored within it [2, 3] — information that is easily revealed; complex pattern-recognition algorithms are required to reveal information that is more deeply hid-

den. Domain knowledge is important in determining search terms and keywords that will retrieve information required from the myriad, heterogeneous data sources built on varying schemas. IR in bioinformatics is, however, compounded by the fact that different research fields often use different and sometimes conflicting terminology for recording and annotating data [84]. Ontologies may serve as a reference framework here, semantically encoding data to reduce ambiguity, leading to improved searching and IR [19, 74], by providing both general and domain-specific semantic knowledge [18]. Visualisations that are mapped to the semantic structure of data provide cues to users in identifying regions most likely to contain information required, and where focused search will be most effective [44]. The ability to classify data based on user-specified criteria aids the location of areas that may contain data of interest. Confidence in the ability to perform effective IR encourages further exploration; coupled with good support for navigation, users are more likely to become immersed in data and continue to obtain a good understanding of data structure.

Text is the simplest method for recording and describing data, and is quick and effective for directed searching and IR using structured, sorted data sets. It is, however, limiting for describing spatio-temporal data such as the anatomy ontologies and gene expression data studied for this thesis. Using images to store the latter allows for a richer representation of data, with a larger number of dimensions and options for encoding data properties. Searching for information within images is, however, far more complex than for simple text [13]; there is the need to provide spatial and sometimes temporal mappings to aid interpretation of image data. One solution is to attach annotation to images to allow textual searching to be performed.

A limitation in text searching is that it reveals information on only search hits, with ranking (often based on automated criteria) as an added extra for some systems. This results in what is described as “*near blind*” searching in [95], associated with a high cognitive load on users. [66] also recognise the benefits in helping users understand how queries are interpreted by a system. Visual representations of query results can be encoded to show (user-specified) degree of relevance of an entire data set to search criteria, so that the distribution of results can be seen both for data satisfying search criteria and for non-search hits. Colour, shade and intensity, and shading in greyscale may be used to encode data properties, and physical distance mapped to semantic similarity to provide intuitive ranking of query results. This helps users to more easily reformulate queries, using visual cues that provide information on areas where data that satisfies users’ requirements is likely to be found.

This can be extended to visualisation of the contents of a data set(s) and the whole process of querying [125, 165]. In an evaluation that compared querying using a direct manipulation interface (DMI) to form-fill and text-based systems [3] found that the DMI resulted in an increase in efficiency and a significantly smaller number of errors. Users’ (subjective) comments revealed that the visual overview of the original data set in addition to visualisation of the query process itself improved IR. Users were more willing to explore the data set using the DMI system because they found errors were easily corrected or

reversed. [162] also obtained similar results in a comparison of text-based to dynamic querying, and with speed of querying significantly faster for the dynamic query interface.

Advances in technology and imaging now make it possible to perform pattern recognition in images and other multimedia data. Visualisation can also be used to create richer and more intuitive interfaces for the query process. Exact and range filters built into sliders may be used for dynamic querying and pre-processing of data, to remove non-relevant data (or noise) and improve ability to retrieve information of interest [125]. Range sliders also provide limits for filtering and querying, preventing input errors and providing clues about information stored in data. Multiple query sliders may be used in conjunction to formulate complex queries, removing the requirement to learn (complex) query syntax. Immediate feedback is obtained for requests made [3], so that it is easy to visualise the effects of slight modifications in query criteria during the process of querying. Queries may be easily modified using intermediate or final query results, or reversed altogether. This intuitive method for IR encourages data exploration [163] and uses incremental learning to obtain a rich understanding of data as it is analysed. Figure 2.19 provides a demonstration of the use of dynamic query sliders in the City'O'Scope data analysis tool.

A useful extension to querying is to provide transparent access to and simplified referencing of external, related data; this may be used to verify and enrich information retrieved [185]. Additionally, formatting output so it can be input into other search and data analysis applications widens the scope of analysis that can be performed.

A challenge in visual, dynamic querying, observed in [3], is visualising non-spatial data such that it allows effective mental models of data structure to be built. Encoding of relationships within data influences IR and the confidence with which users draw conclusions about information content. A second issue, common to interactive visualisation, is obtaining sufficient resources to support effective and timely system response [162].

2.5 Visualisation in the development of data analysis tools

Previous sections discussed the importance of perception in data analysis, with § 2.3.2 looking specifically at the use of metaphors for intuitive, visual analysis. This section continues to examine different techniques built into visualisation tools, some of which are based on or make use of the visual metaphors described.

General-purpose tools often employ a range of simple techniques to generate visual overviews of data, with the ability to perform further analysis in ROIs. Such tools, though often limiting for detailed analysis of complex data, still serve as a guide in the development of more specialised tools [154]. Conversely, domain specific tools have limited applicability to general data analysis [71], focusing instead on providing dedicated analysis and IR for a restricted set of requirements. User needs will determine where general-purpose or specialised visualisation tools, or both in concert, provide optimal analysis [4].

It should be noted that visualisation is important not only as an aid in analysis but

also in the creation of interfaces for data analysis systems [122]. Providing visual front-ends to databases, for instance, aids storage and management of data [71], and allows visual querying to be used for data analysis and IR.

2.5.1 Importance of a modular approach to development

The choice of technique(s) employed for analysis of a data set will influence effectiveness of analysis and reliability of information retrieved for drawing conclusions about data. [71] discuss important considerations in choice of techniques or tools for data analysis:

- user domain knowledge and skill in data analysis
- data type(s) being visualised
- information desired, how it is to be used and what it is to be used for
- scalability of visualisation techniques available.

No matter how well designed, tools developed for a specific target or purpose may not meet the analysis needs of users within a different field of work [39]. Further, working methods are often modified to fit specific tasks and stages of work. The ability to customise tools so that they can be moulded to fit individual users' working methods and often changing requirements improves usability [98].

Development of modular components and systems using an object-oriented approach is ideal as it promotes reusability and extensibility [163, 167]. Powerful tools can be built quickly by integrating different modules and components, allowing multiple analysis techniques to be used in concert [154, 153]. Users are able to concentrate on building tools that perform the analysis required instead of on design and development of their component parts. Choices, however, have to be made between the development of generic components, which are more flexible and reusable, and specialised components which are more efficient at the expense of flexibility. Finally, where it does not limit improvements in analysis techniques it is advisable to adhere to published standards to ensure portability and reusability of components developed.

2.5.2 Existing techniques for data analysis

Data mining

Data mining makes use of search algorithms and pattern recognition to automate IR in structured data sets. Providing visual interfaces to databases eases input, manipulation of data stored and IR. Visualisation of the data mining process further harnesses perceptual ability and augments creative analysis and insight in humans [71, 114, 117]. This is especially useful for analysis of unstructured data, where algorithms fail. Even for structured data providing options for interactive analysis allows humans to identify patterns and/or anomalies in data that automated algorithms might overlook [184].

Visualisation techniques used in data mining include scatter plots, cluster analysis, geometric projection such as multi-dimensional scaling and principal component analysis

(PCA), and intelligent data browsing.

2D and 3D scatter plots

Scatter plots lay out data by mapping semantic similarity to inter-object distance, as illustrated in [68] (see also figure 2.14), who use Sammon's non-linear mapping (NLM) to perform multi-dimensional scaling, mapping high-dimensional data to a 2D layout. Another method for achieving dimensionality reduction is to use PCA [89], which maps dimensions or attributes in data to factors, starting with the *principal factor* or most important attribute, with each subsequent factor describing variability between properties of data elements. A disadvantage associated with PCA is that it may result in a loss in data: attributes considered to have low importance contribute little or nothing to the resultant visualisation.

Scatter plots provide a natural method for semantic clustering, grouping elements based on similarity between data attributes. Alternative perspectives of the same data set are obtained by re-clustering data based on user-specified criteria, revealing different relationships between data elements. Figure 2.8 illustrates clustering of related data in a 3D world generated using a system of forces of repulsion and attraction based on (dis)similarity between data nodes. Colour, intensity, hue, saturation, shading in greyscale, shape and size of data nodes can be used to encode data attributes, providing additional dimensions for describing data above the two or three in which the data is laid out.

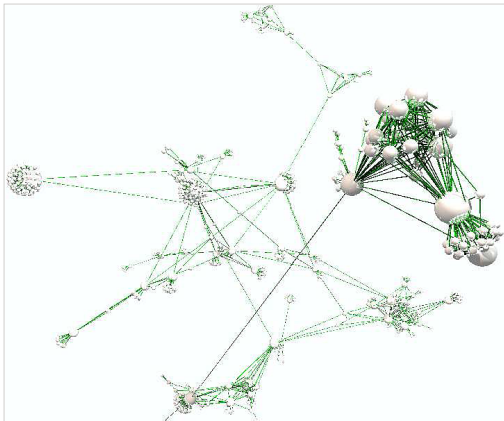


Figure 2.8. *Narcissus* uses forces between objects to lay out data in 3D. When the system comes to equilibrium a scatter plot is drawn in the virtual world created. Two clusters of closely related objects can be seen on the right, with links between objects representing defined relationships. (Image reprinted with permission from [95])

Information murals

Information murals are a pixel-based visualisation technique developed to combat poor scalability in 2D visualisation due to restrictions in display space. This technique generates data overviews with minimal loss of information by using intensity in colour displays or shading in greyscale to encode density of data mapped to a specific pixel location [110].

Hierarchical visualisation

Information or tree maps use a space-filling approach to visualise large, hierarchically structured data sets in 2D [11, 112], to combat the poor utilisation of space common to

hierarchical graphs. The display space is broken up into bounded, rectangular areas, with the amount of space assigned to each node mapped to user-defined levels of significance of data. Colour, intensity and shade can also be used to encode other attributes of data in the space within a bounded rectangle. [112] visualise a directory structure containing six directories and seventeen files using different visualisation techniques, to illustrate the advantages tree maps provide. Three of those visualisations are shown in figure 2.9, comparing the effectiveness of a hierarchical index, a node-link graph and a nested tree graph for displaying the structure and semantic content of (hierarchically structured) data.

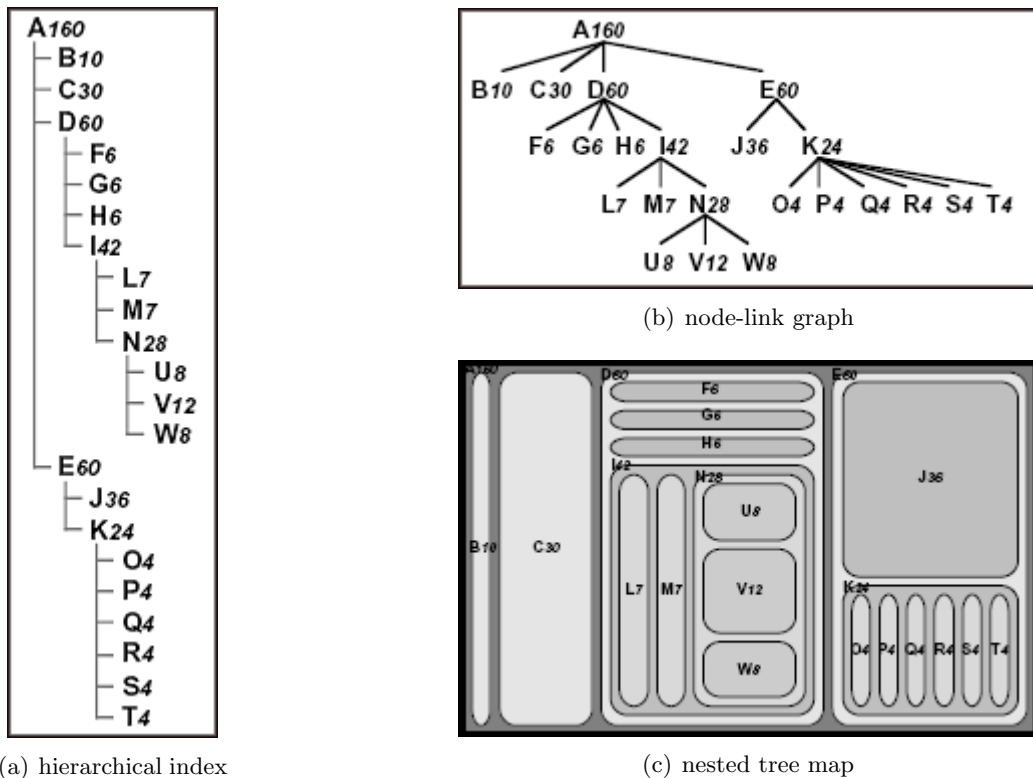


Figure 2.9. Figure 2.9(a) presents the least intuitive visualisation, where a large cognitive effort is required to infer data structure. Figure 2.9(b), though capturing data structure effectively, makes poor use of space. The tree map in figure 2.9(c) provides a compact visualisation that employs physical space as an extra dimension for encoding data attributes. (Images reprinted courtesy of the University of Maryland Human-Computer Interaction Laboratory (HCIL), from [112])

The hierarchical index in figure 2.9(a) provides the least intuitive visualisation — with only 23 nodes and four levels of nesting it is already quite tall, and the data structure is not easily inferred. The hierarchical structure of the data is easily discerned in the node-link graph in figure 2.9(b). However despite the weighted tree that attempts to optimise layout the graph still makes poor use of space. Neither of figures 2.9(a) and 2.9(b) is able to use structure to indicate the size of the file or directory each node represents.

Figure 2.9(c) nests each file or directory within its containing directory, providing, like the node-link graph, intuitive recognition of the data structure. The tree map has further advantages over the other two visualisations in that it maps space assigned to each data

node to its size. Its compact visualisation also makes more optimal use of space. Figure 2.10 presents an even more compact map that makes better use of space available for displaying data, but with the attendant disadvantage of the loss of the visual cues provided by nesting in figure 2.9(c).

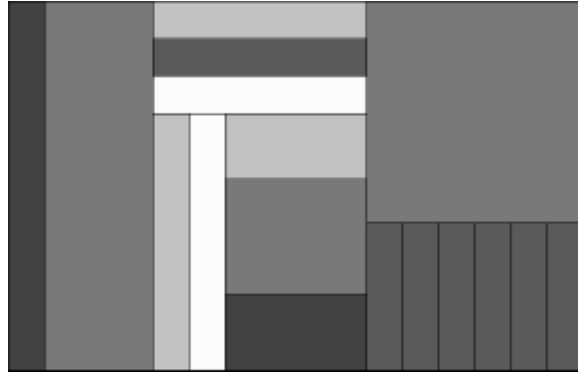


Figure 2.10. This alternative to the tree map in figure 2.9(c) uses shading to encode properties of data elements, but does not nest data. The resulting visualisation is more compact and makes more efficient use of space, at the expense of the loss of the visual cues provided by nesting. (Image reprinted courtesy of HCIL from [112])

A limitation of tree maps is that as nesting grows it becomes increasingly difficult to visualise all levels of a hierarchy in the overview. It becomes necessary to use successive visualisations to display multiple levels in the tree.

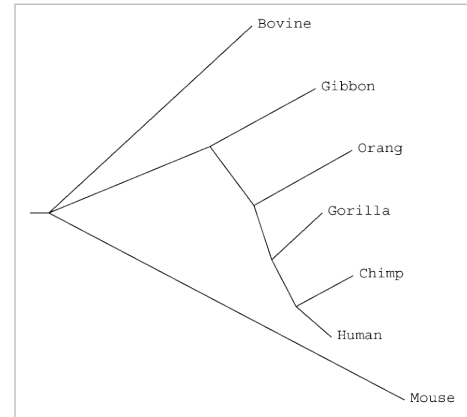
Information cubes provide a 3D extension to 2D tree maps that remove the need for multiple visualisations to display complex nesting. The technique developed by [147] progressively nests sealed containers in enclosing outer containers, to visualise nesting in a hierarchy. Information cubes take advantage of the larger amount of space available in 3D to generate compact visualisations of large, complex data sets.

The outermost container is transparent, so that containers nested within it can be seen along with the text labels that describe their contents. As one descends the tree the level of transparency of containers and detail in annotation decreases, to reduce complexity in the overview. Level of detail in ROIs, however, varies with the proximity of data to the viewpoint, as users navigate through the data. Shape and size of containers, colour and transparency are used to encode data attributes and the importance of data, based on user-defined criteria. Clustering places semantically related data within the same container or in close proximity. Rotation and translation are used to bring ROIs closer to the viewpoint to aid detailed analysis.

Tree graphs provide a useful abstraction for representing the structure of hierarchical data. Data is stored in nodes and leaves in the tree, and relationships in links between nodes. Clustering is obtained by folding the tree at a node, especially useful for managing occlusion, which causes significant problems in hierarchical graphs. [98], for instance, set a

in a data set, while data nodes and leaves represent individual proteins, genes, organs or organisms. Edges between nodes contain relationships in data [88].

Figure 2.13. A cladogram drawn using the web service for the visualisation tool *Phylodendron*, to show evolutionary divergence between seven mammals. (Image created using Phylodendron’s web service at: <http://iubio.bio.indiana.edu/treeapp/treeprint-form.html> (last viewed Jul 2006), and a sample data set from <http://iubio.bio.indiana.edu/treeapp/treeprint-sample2.html>)



Dendrograms, which are only useful for visualisation of hierarchical data [89], are well suited to multi-dimensional, evolutionary data, with its inherently hierarchical structure. Dendrograms lend themselves well to interactive, exploratory visualisation and data analysis, and provide an intuitive method for classification by (hierarchically) clustering like data. [89] provide a demonstration of the use of dendrograms for hierarchical clustering of gene expression data.

One limitation of dendrograms, however, and common to tree graphs, is poor visualisation of horizontal relationships — inter-object distances along the same level have no meaning [68]; arbitrary ordering of nodes in space means that physical separation between nodes may not map to semantic similarity. Figure 2.14 compares a scatter plot to the equivalent dendrogram, illustrating differences in semantic meaning that may be inferred from node layout in the two visualisations.

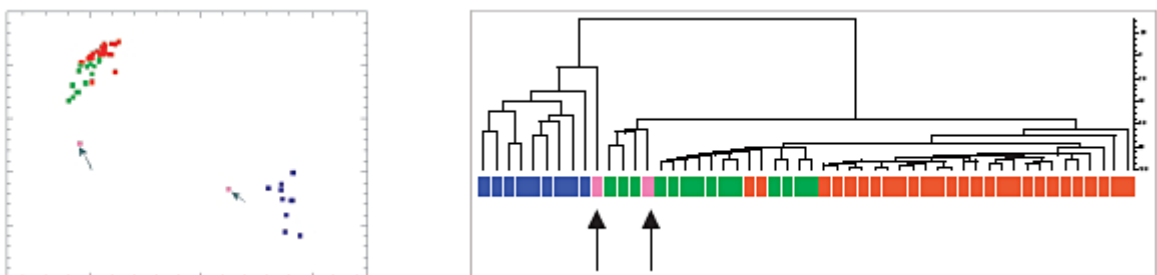


Figure 2.14. [68] use Sammon’s NLM to lay out a scatter plot that encodes gene expression data using colour. Three distinct clusters are seen, with a close relationship between the gene expression data in green and red, and the blue lying some distance away. Two isolated points in pink highlight anomalies in the data. Although clustering occurs in the equivalent dendrogram, inter-object distance and relative position of clusters do not map to (dis)similarity in the data. Information on ancestry may however be derived from branching in the tree. (Images reprinted with permission from [68])

Isomorphism also presents a problem in the use of dendrograms: misclassification of similar data may occur when clustering algorithms used for automated classification fail, resulting in semantically related data being assigned to different clusters [160, 88]. A simple

solution to this is to provide interactivity that allows perception in humans to be used to identify and correct errors in automated classification and clustering [89].

Another limitation of dendrograms and cladograms is their poor scalability [88]; few applications that make use of hierarchical trees are able to support visualisation of more than a few hundred to a thousand nodes and edges before severe occlusion renders the overview unusable.

Cone trees are an interactive visualisation technique developed by [152] (see also [151]), that takes advantage of the added dimension of depth in 3D to improve use of screen space for the analysis of large, hierarchically structured data sets. Starting with the root node at the top of the volume occupied, the cone tree visualisation successively draws sub-trees with each parent node at the apex of a cone, and its children uniformly distributed along the circumference of its base.

Depth cues in the 3D world are augmented with colour coding and lighting. [152] also use light to throw shadows of the cone trees generated to the base of the structure, to provide additional visual cues that aid understanding of the structure of the hierarchy formed and clusters that occur within it. Natural perspective in 3D increases magnification of data nodes in the visual structure with proximity to the viewpoint.

The 3D structure, however, suffers from the inherent problem of occlusion of more distant data elements. Two methods are used to reduce this: cones drawn are semi-transparent, so that objects lying behind them are still visible. The second solution uses *cam* trees, a rotation of the cone tree visualisation that draws trees along the horizontal axis. Node labels can be drawn for individual data elements with a significantly lower level of overlap in *cam* trees than occurs for the vertical layout. Finally, the data structure may be rotated to bring data of interest to the viewpoint.

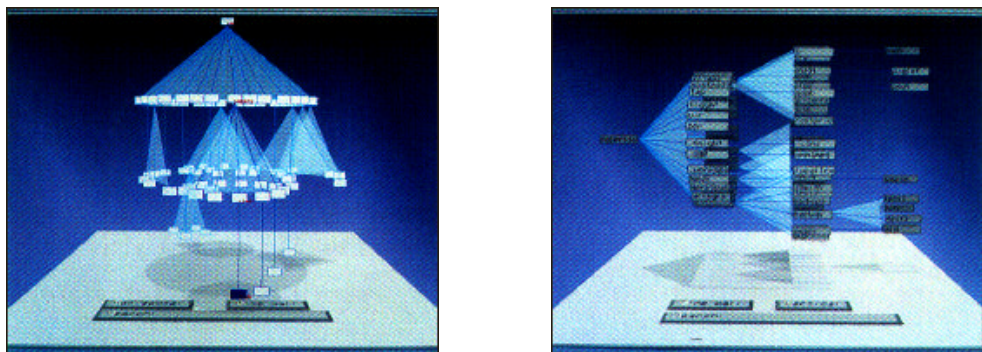


Figure 2.15. A cone tree [152] is shown on the left, with its equivalent *cam* tree on the right. Both visualisations show the additional visual cues provided by the shadows that are cast onto the base from the light above each structure.
(Images reprinted with permission from [152])

Using 3D allows more data nodes to be displayed than for the equivalent 2D visualisation; it is not possible to draw equivalent visualisations in 2D for the cone trees system that are able to store the same amount of information at the same magnification and screen size.

However, as is common to hierarchical visualisation, severe occlusion still occurs beyond a relatively small number of nodes; the cone trees system is only able to display about 1000 nodes before usability of the overview is degraded by occlusion.

Unlike most visualisation techniques developed first for 2D and then extended to 3D, cone trees were originally developed as a 3D visualisation technique. This eliminates distortion and other visual defects that occur when projecting what are essentially 2D visualisations into 3D space. [116], however, argue that (3D) cone trees present several problems for visualisation, especially when projected onto a 2D display. They observe that the hierarchical structure of the data is easily recognised when viewed from above, but from the side or within the visualisation occlusion of more distant nodes obscures its overall structure. Loss of context occurs when immersed within the data in order to analyse ROIs, or when moving between different layers of the tree, resulting in an erosion of users' mental models of the data structure. Another problem observed, common to hierarchical graphs, is that physical distance between node pairs is not meaningful, so that it cannot be used to interpret semantic distance between data elements.

Focus+Context Techniques

Perspective and hyperbolic projections or fish-eye views provide a wide field of view, with maximum magnification at the focus, which falls away progressively with distance [120, 159]. The projection makes use of the exponential increase in space in hyperbolic layouts to (re)draw ROIs at higher magnification, reducing occlusion without sacrificing the context of the overview.

This is especially useful for visual analysis of large data sets, where retaining context helps to reduce disorientation during navigation. However, distortion in the visual structures generated and constantly changing focus in hyperbolic layouts may prevent users from forming consistent mental models of data structure [127, 159], making it more difficult to recognise relationships within data. One example of an application that makes use of a perspective projection is the cone tree system developed by [152].

Perspective walls use natural perspective in 3D to provide an F+C system for visualising linearly structured data [151]. Figure 2.16 shows the 3D wall formed by folding away the context from the central section and focus of the data. The focus can be changed interactively by dragging items to the centre of the wall; this results in relative relocation of other data closer to or away from the viewpoint, with a respective increase or decrease in magnification of objects lying on the wall.

Shading in greyscale or intensity for colour displays may be used to enhance the perception of depth, increasing the ability to recognise and understand relationships within data. Perspective walls may be used as timelines, plotting changes in the properties of data from one end of the wall (in time) to the other.

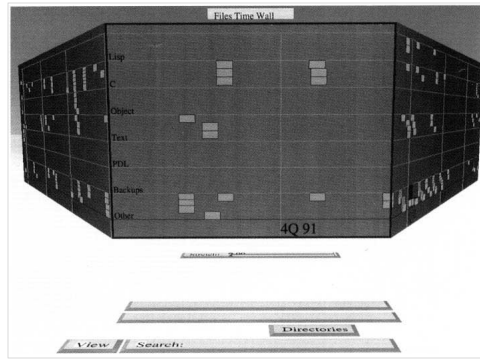


Figure 2.16. [151] use a perspective wall to visualise the structure of a file system. Items of interest can be taken off the wall and placed in front of the user, to allow examination in detail. (Image reprinted with permission from [151])

Combined with a *rubber* metaphor that stretches the wall, the level of detail and consequently occlusion, may be managed interactively [151].

Portals

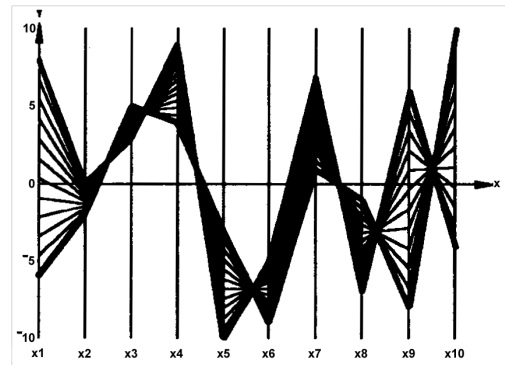
[138] illustrate how *portals* can be used to provide intuitive navigation between widely separated locations in a data set using *Pad*, a visualisation system that places data on an infinite 2D surface. [140] use a similar system to aid comparison of widely separated data elements, by providing a magnified view onto one ROI from another. [138] also extend the metaphor to generate semantic filters that alter physical properties used to encode data, to provide different perspectives of ROIs viewed through *portal filters*.

Parallel co-ordinates

In order to map experience in the real world to data analysis, graphical analysis is normally limited to a maximum of three (physical) dimensions, with data drawn in space along mutually orthogonal axes. This requires dimensionality reduction for high-dimensional data, which often results in a loss of information. The parallel co-ordinates technique provides a graphical method for exploratory visual analysis of multi-dimensional data without the need for data reduction: it draws polygonal lines by joining points on a set of equidistant, parallel axes lying on a 2D plane, each representing an attribute of the nodes in a data set [69, 108, 109]. Physical limits to the number of dimensions available for describing data attributes are removed.

Colour, thickness and transparency of lines between axes are used to highlight data of interest. Clustering is obtained where high similarity occurs in values for data attributes, as is seen in figure 2.17, and outliers are easily identified. The parallel co-ordinates technique is also useful for analysing temporal data, where the parallel axes serve as a timeline.

Figure 2.17. [107] demonstrate the use of parallel coordinates for visualisation of multi-dimensional data.
(Image reprinted with permission from [107])



Beyond a relatively small number of dimensions it becomes necessary to scroll to view different attributes. It is also difficult to compare attributes on widely separated axes. The latter problem may be resolved by rearranging axes to bring selected attributes closer together, to ease comparison of data.

[69] extends the parallel co-ordinates technique to 3D (see figure 2.18), drawing the parallel axes as 2D planes, with each polyline in an individual plane perpendicular to the *axes*. The 3D visualisation provides more degrees of freedom to the user for examining its structure: the visualisation may be rotated, allowing viewing from arbitrary angles, in addition to the translation available in 2D. Facing the parallel *axes* at right angles results in the view obtained for the equivalent structure drawn in 2D. Viewed from above a major benefit of 3D can be seen: elimination of the crossing of lines that occurs in 2D. Each plane representing a specific attribute contains a scatter plot; if data elements are arranged such that physical distance maps to semantic similarity a layout of data nodes could be obtained that provides additional visual cues for analysis.

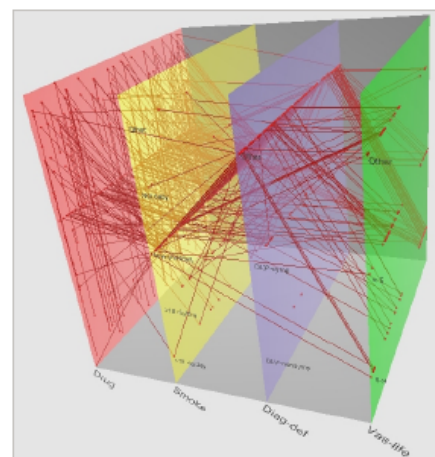


Figure 2.18. [69] illustrates use of the parallel co-ordinates visualisation technique in 3D space to compare medical records.
(Image reprinted with permission from [69])

[26] combine multiple visualisation techniques to obtain the *City'O'Scope*² (last viewed Jul 2006) visual data analysis tool shown in figure 2.19. A set of parallel co-ordinates use colour and thickness to highlight polylines representing data elements of interest, and clus-

²More information on *City'O'Scope* is available on the Macrofocus web site at: <http://www.macrofocus.com>

tering is obtained along each axis where a large number of elements display high similarity in attributes. Dynamic query sliders are used to construct composite queries that filter out data not satisfying search criteria. This is reflected in the corresponding dynamic scatter plot where clustering also reveals similarity in data attributes, based on criteria used to lay out data. The fourth component in the visualisation tool is a world map that highlights cities meeting query criteria. A fish-eye view magnifies ROIs on the map and brings them to the centre and focus.

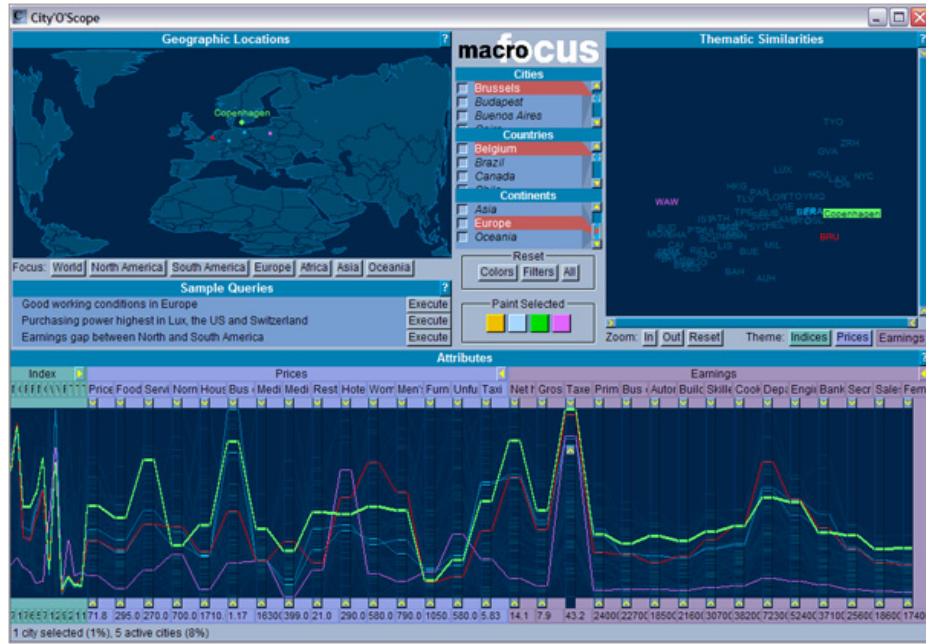


Figure 2.19. *City'O'Scope*, described in [26], provides interactive visualisation of economic data for a number of cities in the world using multiple visual analysis techniques in concert. The snapshot, printed with permission from the application, shows use of a demonstration version of *City'O'Scope*.

Virtual reality

Harnessing metaphors in real life to generate visualisations aids navigation in virtual worlds, and provides intuitive data exploration and analysis. *Walk-throughs* may be supplemented with enhanced metaphors that allow users to *fly over* or *through* data, jump through *portals* and interactively modify physical attributes of data using *magic wands* while immersed in the data. In-built support for navigation in virtual reality (VR) includes multiple viewpoints or cameras and landmarks [160]; there are also the larger number of degrees of freedom available in 3D for navigation. Optimal use of VR, however, requires additional support for software and hardware that is not commonly available in the average working environment using personal computers (PCs).

2.5.3 Interactivity and animation

Giving users control over how data analysis is performed gives more confidence in results obtained. Providing options for interactivity allows users to work more directly with data so that a more intimate knowledge of data is obtained [48, 71]. Domain knowledge can be brought to data analysis, helping to validate results of automated analysis and clarify ambiguous or incomplete information retrieved. This is especially useful for irregularly structured data, which automated algorithms deal poorly with [2, 72].

Visual, interactive analysis provides a spatial extension to limited human short term memory, benefiting from perception which reduces cognitive load in analysis, and leading to better understanding of data [39, 71]. Clustering, grouping or classification of data based on user-specified criteria provide alternative perspectives [117] and reveal variation in interaction within data [69]. Level of detail, both physical and semantic, may be varied as users move closer to ROIs or draw away to obtain a wider view of data [165].

Complex queries [74] do not always retrieve information desired. In such cases interactivity combined with visualisation provides an advantage. The query process can be visualised, and queries refined or extended based on intermediate or final results, to obtain near-optimal or alternative solutions. Incremental querying, such as described in [152, 162], can be used to reveal search hits as they are found, instead of waiting for all data satisfying search criteria to be retrieved. This allows errors in query terms or anomalies in data to be identified quickly, so that modifications to queries can be made during the search process, increasing the probability that final results will contain information required [125].

Quick, simple identification of errors, usable system feedback and support for error recovery all have significant impact on the willingness of users to perform exploratory analysis. The ability to return to previous states is also important, especially for navigation through complex data, to re-orient users who get lost within data. The availability of history sessions and the ability to place markers within data and save system state also contribute to interactive, incremental or continuous visual data analysis [165].

The support required for interactive visualisations is naturally higher than that for static visualisations [163]. Larger amounts of computational resources are required for animation, to redraw visualisations in real time. It is sometimes necessary to sacrifice high-end graphics and resolution for improved response, especially during navigation and exploratory analysis [122]. Sudden jumps between ROIs or large changes in the structure of visualisations in response to user actions may result in disorientation [98, 159], while users perform the cognitive transition from one perspective of a visualisation to another. However there is evidence that suggests that smooth animation as a layout is updated may help users form good mental models of data structure [39, 48, 74, 151]; the cognitive effort required to identify relationships among different elements is transferred to the more intuitive perceptual system [152]. Browsing and navigation through data improves, users

are able to move more easily between ROIs, progressively building an understanding of interaction between different data elements and the overall structure of data [21].

However restrictions may be required to prevent users from distorting visual structures to the extent that they are no longer meaningful, and mental models created of the data structure are destroyed [39, 48, 96]. Limiting degrees of freedom for navigation available to users, especially in 3D, also reduces the occurrence of disorientation.

Providing meaningful presentation of results that are understood by audiences with different skills and research backgrounds is another challenge faced in data analysis. Interactive generation of visualisations may provide a useful option for presenting the results of analysis.

2.5.4 Limitations in visual analysis

Despite improvements in technology, restrictions in screen size and resolution, the graphics capability of systems available both for generating and viewing visualisations, and computing power still remain significant limitations to effective visual data analysis [116]. The scalability of techniques developed for visual analysis is also a problem; intuitiveness and manageability of visualisations decrease significantly as data set size and dimensionality increase.

2.6 2D vs 3D

Visual data analysis is increasingly being performed on PCs, with data drawn on 2D displays. This naturally suggests that the ideal number of dimensions in which to present data would be two. However, except for fairly small data sets (containing only a few hundred data elements) occlusion in 2D visualisations often limits the usability of data overviews. An extension to 3D provides extra space for holding data [96, 127], helping to reduce the occlusion that occurs in the limited space available in 2D.

A significant advantage in 3D is the added spatial dimension that allows a transfer of the cognitive effort required for data analysis to more intuitive spatial memory [51]. The extra space available in 3D, due to depth, results in higher density of data, increasing the ability to uncover information required and the effectiveness of analysis [151]. [152], for instance, found that it was not possible to generate equivalent hierarchical structures in 2D using the same amount of screen space as they achieved with their 3D cone trees visualisation technique. It would be necessary to draw the 2D structure at significantly lower magnification, or require scrolling to view the entire structure at the same magnification, with an attendant loss of context (and increase in cognitive memory load). It may however be argued that the need to rotate 3D structures to view distant data elements occluded by others closer to the viewpoint is analogous to scrolling in 2D.

Objects that more closely approach equivalents in the real world may be used to generate virtual worlds in 3D [69, 99], allowing use of prior knowledge in navigating through real life to aid navigation through complex, interactive systems, and reduce the cognitive burden

on users [59]. A larger number of degrees of freedom are available for navigation in 3D, including rotation, which is especially useful for bringing more distant data elements closer to the viewpoint. An added benefit is that users are able to immerse themselves into data, forming an understanding of the structure of the information space and relationships within data as they move through the visual structures.

Another advantage is that 3D projection provides natural perspective, increasing magnification as objects approach the viewpoint, so that more detail is seen in ROIs without losing the context of surrounding data [69]. This provides some of the benefits of the wide angle lens used in hyperbolic layouts without the distortion that is an artefact of this technique, and also depth cues, which aid visual analysis.

3D visualisation comes with its own problems, however, chief of which are disorientation during navigation and occlusion of more distant elements [69, 163, 164]. Additionally, [51] found that the larger number of degrees of freedom for exploration may hinder rather than aid users in locating information, where higher density of data produced more clutter rather than more information. [51] note however that their experiments used a fairly small data set, and that comparison of significantly denser displays for different dimensions may produce different results.

Reduction to 2.n dimensions (n between 0 and 5) makes use of the landscape metaphor, eliminating disorientation due to complete immersion in 3D [39]. Users are able to remove themselves to a plane above the visualisation, to *fly over* data and obtain an overview that provides context for analysing ROIs. Fog and lighting may be used to enhance the simulation of the (virtual) landscape, employing a geographical metaphor to aid navigation and location of data of interest. Placing easily identifiable and recognisable markers and multiple cameras or viewpoints in data helps to re-orient users who get lost, by providing easily reachable landmarks [69]. [182] found that provision of landmarks as for wayfinding in the real world aids navigation through especially large data sets, where it is difficult to obtain an overview of data, preventing disorientation by helping users incrementally build mental maps of overall information space. [99, 151] discuss additionally, the use of portals or magic doors, history and undo facilities to help manage difficulty in navigation and exploration in 3D.

Effective generation of visualisations within the limitations of software and hardware available, that support smooth, controlled motion of users and objects within 3D worlds, and that provide effective visual cues, is especially a challenge for desktop applications, which make use of 2D for both user input and data display. Mapping a 3D representation to 2D screens may result in some distortion [99], increasing the probability of disorientation occurring and the potential for misinterpretation of data structure. More support for both hardware and software is required for generating and visualising 3D computer graphics. Fully immersive 3D systems with haptic feedback and stereoscopic displays may be used to provide a better illusion of (3D) space, and provide additional physical and visual cues that enhance exploration of and navigation through virtual worlds. This may help to make

up for the peripheral vision unconsciously used in the real world to obtain contextual information, which is lost in the projection of 3D worlds onto desktop environments. Even though advances in technology have significantly reduced the cost of resources required for computer-based solutions while enabling improved analysis, the higher cost of hardware and software required for effective generation and display of 3D worlds still restricts optimal use of 3D for the average user [99]. Also, apart from the higher financial cost associated with such systems, most analysis is performed using PCs in office environments, where such solutions are not practical.

Several factors come into play when making a decision to use 2D or 3D for visualisation of abstract data. 3D space provides advantages over 2D for large amounts of high-dimensional data. However anecdotal evidence suggests that visual representation of data in 2D, using less complex techniques for navigation and exploration of data, is easier for users to learn to use on the 2D displays normally available in the workplace. The larger number of degrees of freedom in 3D, though helping to move around objects to manage occlusion of more distant data elements, also contribute to disorientation.

[66, 69, 91, 161] found that tools that do not match user ability and normal working methods, or provide options that do not satisfy users' information needs are unlikely to be used; [6] found that poor mapping between tools and user ability may reduce performance in analysis. [39, 44, 50] additionally discuss the importance of mapping semantic meaning and the structure of data to visualisations generated, and to users' specific and changing information requirements and working environments. [6, 51], among others found that effectiveness of 2D and 3D displays varies depending on the tasks being performed; [44] also stress the importance of recognising differences in user ability and domain knowledge, and experience in the use of computer-based tools, and the impact these have on ability to make use of visual analysis tools. Research shows that varying degrees of spatial ability or awareness in individuals influences their ability to navigate effectively through higher-dimensional environments [6, 44], their understanding of visualisations and the information they contain [181], as well as their general use of graphical user interfaces (GUIs).

Tools that provide very effective data analysis may still be rejected if they are difficult to learn to use or have poor response. Design and development of tools that cater to the individual needs of users with what may be wide variations in ability, domain knowledge and backgrounds, is a challenge especially for interactive analysis. Structured evaluations of the visualisation browsers developed for this project (see chapters 7 and 9) found that despite greater difficulty navigating through the 3D information space, users on average preferred the 3D to the 2D browser. Users recorded improved ability to approach data of interest and easier interpretation of colour coding in 3D. 3D also enabled simultaneous analysis of multiple data sets; space limitations in 2D meant the same could not be achieved using a single 2D window. A factor that may have biased opinions on usability was the exponential increase in delay in system response that occurred in the 2D browser with increase in data

load for the first structured evaluation; differences in delay in the 3D system with data load were negligible (refer § 7.2). However this would have been tempered by inherent complexity in especially navigation in 3D browser, exhibited especially during the second evaluation as more complex tasks were performed.

2.7 Analysing data over a network

Computer networks and the Internet ease data sharing and exchange, especially for users in widely separated geographical locations, [90], due largely to a set of standards for data transfer and presentation that are widely adhered to. With sufficient bandwidth and fast networks it is possible to perform data analysis online, using software and storage facilities in remote locations. This removes the need to download and store large data sets and applications [45], which is especially useful where local resources are limited. Added bonuses are the removal of the burden of data management from users and improved ability for collaborative work among geographically dispersed users. Data updates can be obtained in real time, allowing continuous, incremental analysis to be performed. Making tools available online also eliminates the supply chain in the distribution process, cutting financial cost and time, so that software releases and updates are available to users immediately they are ready to be deployed.

However interfaces that can be developed for web applications are more restrictive than those for equivalent standalone applications. The usability of interfaces for web-based applications still has a high impact on whether or not tools are used. An advantage in these tools is that where they are properly designed, web applications allow users to take advantage of prior knowledge in using web sites to learn how to use online tools quickly. However differences in the look and feel of graphical components for web and standalone tools reduce the transferability of learning between the two application types.

Other usability problems typical to web applications include long response times due to large downloads and slow networks, and security issues that restrict use of applications [35]. Requiring users to download and install software to make use of web applications creates additional problems, especially when they are non-standard.

2.8 Summary

This chapter examined the capability of humans for complex data analysis, looking at methods that can be harnessed to provide intuitive analysis. Metaphors commonly used to generate visualisations were reviewed, along with visual analysis techniques building on these metaphors, to identify the merits and limitations associated with each.

The advantages provided by each of 2D and 3D were discussed, looking at additional factors that influence users in the choice and use of data analysis tools. The chapter concluded with a look at the dissemination and use of data analysis tools over a network.

Chapter 3 continues to examine graph visualisation, as an option for visual analysis of the hierarchically structured ontology data being studied.

Chapter 3

Graph visualisation

Graphs provide a simple but effective method for visualising the structure and content of data [49]; visualisation allows encoding of quantitative data so that it can be presented in qualitative form, transferring cognitive effort required for data analysis to more intuitive perception. One of the simplest implementations of graph visualisation is the node-link graph commonly used to display relationships within data [5, 99], in which data elements are represented by nodes, and relationships between nodes by edges or links. Colour, saturation, shape and size of nodes and links may be used to encode different properties of data, and map relevance or importance of data nodes [96].

The tree graph is a specialisation of the node-link graph that lays out data using a hierarchical structure. A disproportionate increase in occlusion in node-link graphs with data set size [128], however, results in difficulty distinguishing data nodes and navigating through data. Abstraction that employs clustering of like data helps to manage complexity and occlusion [74, 117], and uncovers relationships within data based on alternative classification criteria. Hierarchical clustering also aids navigation, hiding more distant and less immediately relevant data, and revealing greater semantic and physical detail as one approaches data of interest.

3.1 Common applications of graph visualisation

Graph visualisation is widely applied to work in daily life. Example application areas include the design of interfaces for file and document management systems, hypermedia systems, organisational charts, web site maps and for browsing history lists on the web. Graph visualisation is also useful for examining semantic maps and networks, drawing genetic maps, entity relationship (ER), state transition (STDs) and data flow (DFDs) diagrams [56].

Hierarchical graph visualisation is useful for mapping paths through data, with its ability to fold away detail in more distant areas of a structure, revealing more information and additional alternatives for navigation as users approach an ROI. Hypertext and hypermedia

systems are good examples of data sets with a large amount of interlinking among data nodes, often leading to disorientation during navigation. Figure 3.1 shows how a node-link graph is used to map the structure of a web site, based on interlinking between documents. Figure 3.2 shows alternative layouts for the defined navigation structure of the same site (which does not necessarily map to the physical structure).

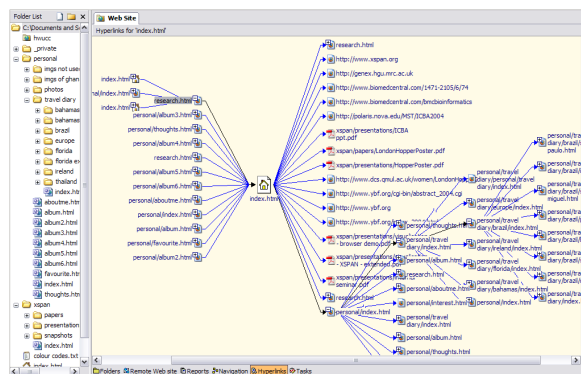


Figure 3.1. Hyperlinks between documents in a web site are displayed using a node-link graph in Microsoft Frontpage®. Because links are uni-directional nodes may appear multiple times; to reduce the occlusion this causes only one sub-tree per level can be expanded at a time, with this path traced in black. The path from the node with the focus to its parent is highlighted in red.

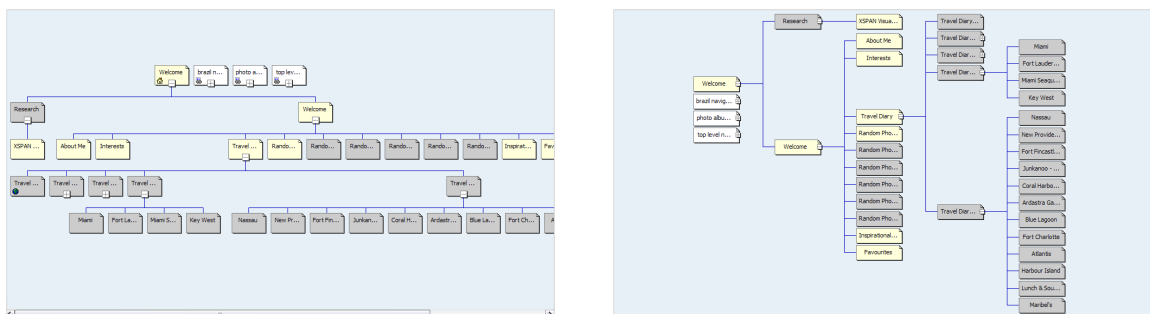


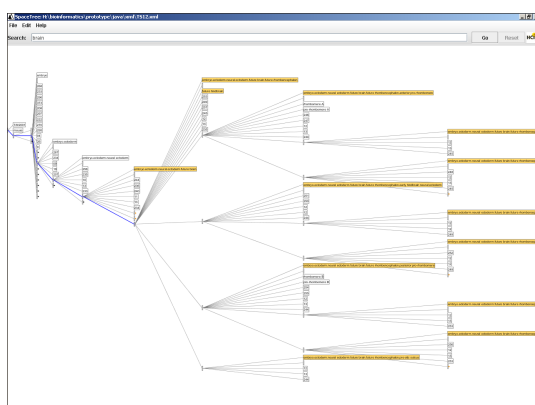
Figure 3.2. The (top-down) layout is shown on the left for the graph that stores the defined navigation structure for the web site in figure 3.1. Sub-trees collapsed to obtain a more compact structure are marked with a closed stub on each composite node. The equivalent horizontal layout is shown on the right, displaying the same number of nodes in a smaller amount of space than is required for the vertical layout. Both graphs still make poor use of space in the display.

3.2 A review of existing graph visualisation tools

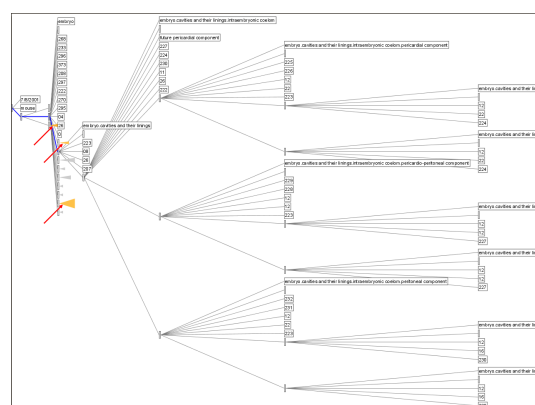
The following sections take a look at a sample of graph visualisation tools, both general-purpose and specifically designed for bioinformatics data analysis, identifying the main strengths and limitations of each.

SpaceTree¹

SpaceTree [139], developed using Java, extends the classic node-link graph using the zoomable interface of *Jazz*² and variable cameras to provide dynamic visualisation of hierarchical data. Graphs generated are automatically modified to suppress data outside ROIs, providing a larger amount of physical space in which to perform detailed analysis of data of interest. Context is maintained by attaching visual and textual cues to composite nodes, to provide information on the sub-trees they contain, as illustrated in figure 3.3(b). The path from the node with the focus to the root is highlighted, to help provide context and a sense of direction to users.



(a) Using the search function in *SpaceTree* to highlight nodes containing the term *brain*.



(b) A search for *membrane* highlights the icons containing the sub-tree where it can be found.

Figure 3.3. A demonstration version of *SpaceTree*, developed by [139], is used to visualise the EMAP XML file representing Theiler Stage (TS) 12 of development of the mouse embryo, containing 199 anatomy components, each with sub-elements describing component properties. *SpaceTree* provides detail for the region with the focus and for surrounding data, folding away other sub-trees and reducing the need to scroll or pan through the graph.

(The version of *SpaceTree* used to visualise the data may be downloaded from: <http://www.cs.umd.edu/hcil/spacetree>. Images printed with permission of HCIL.)

uDraw(Graph)³

Formerly known as *daVinci Presenter* [76], *uDraw(Graph)* provides an interface that visualises relationships within data using hierarchical graphs. *uDraw(Graph)* is also designed to work as a plug-in within other (data analysis) packages. Useful features in *uDraw(Graph)* are support for incremental layout and interactive modification of the visualisations generated. Encoding of data attributes is obtained using shape, size and colour of nodes and links. The *uDraw(Graph)* interface is developed using Tcl/Tk.

¹See the SpaceTree web pages at HCIL: <http://www.cs.umd.edu/hcil/spacetree>

²Other applications of Jazz can be found at <http://www.cs.umd.edu/hcil/piccolo/applications/index.shtml#jazz>. See also Piccolo, which succeeded Jazz, at <http://www.cs.umd.edu/hcil/piccolo>

³See the uDraw(Graph) web site at: <http://www.informatik.uni-bremen.de/uDrawGraph/en/>

HyperGraph⁴

HyperGraph is an interactive, Java-based application that makes use of a hyperbolic layout in 2D to draw large trees. The hyperbolic layout has two main advantages: it is able to visualise a much larger number of nodes than the equivalent Cartesian layout, and it provides an F+C system that provides greater magnification to data in ROIs, as illustrated in figure 3.4.



Figure 3.4. The two snapshots show navigation through a wiki: black nodes represent leaves in the tree, those in blue may contain a collapsed sub-tree, and red nodes highlight errors in the data. Significantly lower clutter is seen in the region surrounding the node with the focus, *XML*, encircled in green. Even though this node lies in the same general area there are significant differences in the layout of the tree, a problem common to hyperbolic layouts and which prevents a consistent mental model of data structure from being formed. (The sample data set used and images printed with permission from the *HyperGraph* web site at: <http://hypergraph.sourceforge.net/item562489632.html>.)

VRMLgraph⁵

VRMLgraph uses a Java application to write the structure of 3D node-link graphs to VRML (Virtual Reality Modelling Language) files (see figure 3.5). The main advantage in *VRMLgraph* is that the visualisations generated are able to take advantage of the additional degrees of freedom available in 3D and in-built functionality for navigation in VRML browsers, to aid exploration of and navigation through the data being analysed.

Walrus⁶

Walrus [106] is a visualisation tool developed using Java3D that draws very large directed graphs in 3D hyperbolic space. *Walrus* is able to display comfortably hundreds of thousands of nodes before occlusion becomes a problem. The hyperbolic layout, shown in figure 3.6, also provides a changeable focus that allows analysis of ROIs within the context of the overview.

⁴See the HyperGraph web site at: <http://hypergraph.sourceforge.net>

⁵See the VRMLgraph web site at: <http://vrmlgraph.i-scream.org.uk>

⁶See the Walrus visualization tool web site at: <http://www.caida.org/tools/visualization/walrus>

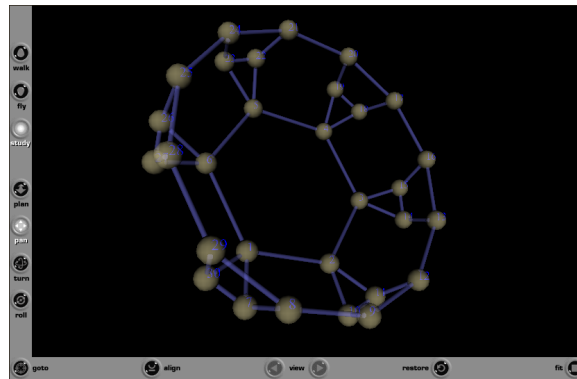
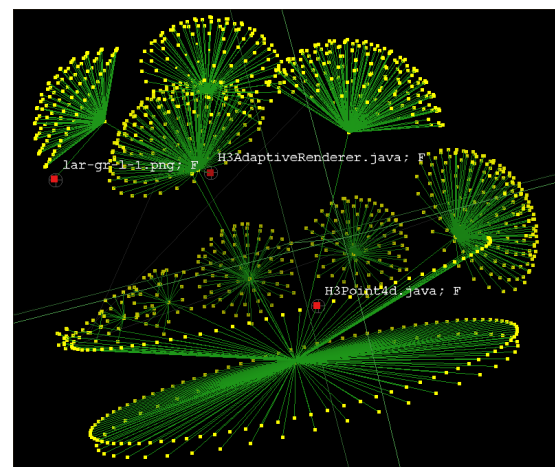


Figure 3.5. A sample data set used to generate a 3D node-link graph using *VRMLgraph*

Figure 3.6. The *Walrus* application described in [106] is used to visualise the directory structure of the folder that contains its source files, displaying 12 attributes in over 1100 nodes and links. The view displayed zooms in to the graph and highlights three nodes, displaying their base names and their (common) root.

(Image printed with permission using a sample file and the Walrus application downloaded from: <http://www.caida.org/tools/visualization/walrus>)



Phylogenetic Tree Drawing Tools

Although the analysis for the research areas in EMAP and XSPAN are not looking at phylogeny, the hierarchical visualisation techniques used in drawing phylogenetic trees may still be useful for generating overviews of the data sets being analysed. A large number of tools for drawing phylogenetic trees exist, including *PHYLIP* (described in § 4.2), *TreeView*⁷, *TreeJuxtaposer*, the *ATV Viewer*⁸ and *BioLayout*⁹. The main features of the last three tools are summarised below.

TreeJuxtaposer, developed by [130] using Java and OpenGL with GL4Java¹⁰ bindings, employs variation in colour, shading and saturation to encode ROIs in phylogenetic trees, highlighting equivalence across multiple trees based on structural similarity between nodes. *TreeJuxtaposer* maintains an overview containing all data elements rather than more commonly used data abstraction that minimises the number of data elements drawn to the screen. A *rubber sheet* metaphor is used to provide F+C for ROIs, allowing users to stretch regions in a tree to provide greater magnification, and/or drag less important data away

⁷See the TreeView web page at: <http://taxonomy.zoology.gla.ac.uk/rod/treeview.html>

⁸See the ATV web pages at: <http://www.genetics.wustl.edu/eddy/atv>

⁹See the BioLayout web pages at: <http://cgg.ebi.ac.uk/services/biolayout>

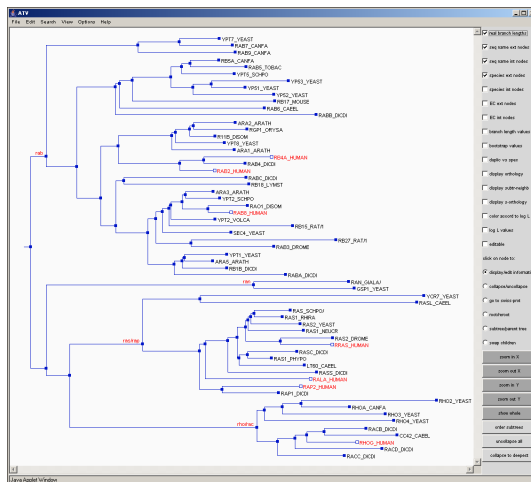
¹⁰See the *OpenGL for Java* web pages at: <http://gl4java.sourceforge.net/>

from the focus.

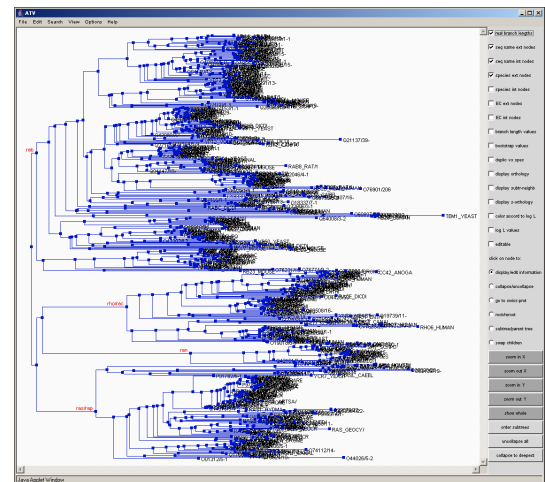
A major advantage of *TreeJuxtaposer* is design for scalability: a single tree of about half a million nodes may be visualised, while maintaining usable system response. Multiple trees with smaller sizes may also be displayed simultaneously without sacrificing system response.

The ATV Viewer⁸, developed by [186] using Java, allows interactive editing of underlying data, (re)drawing and analysis of phylogenetic trees. Options for editing layout of trees include re-rooting, collapsing of sub-trees into parent nodes, and re-ordering of child nodes. Variation in lengths of links may be used to encode data, or the tree may be arranged to align all leaves, removing semantic meaning attached to links. Colour may also be used to encode data properties, and functionality is provided for string searching on element names and identifiers (IDs).

ATV Viewer may be run as a standalone application or as an applet. Snapshots of the applet version of *ATV Viewer* are shown in figure 3.7, comparing the data set for the *ras* full tree with the *seed tree* downloaded from the *ATV Viewer* web site.



(a) A search for the string *human* highlights labels hits in the graph in red.

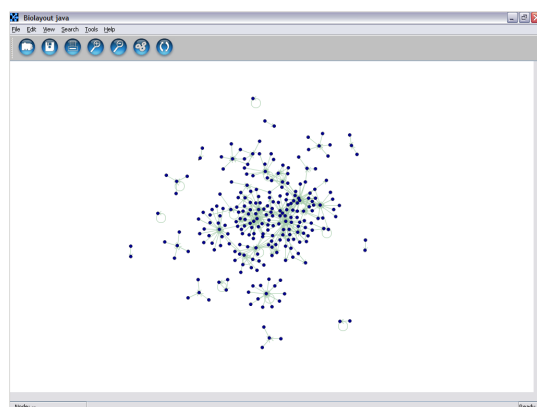


(b) Significant occlusion especially due to node labels occurs for the larger data set.

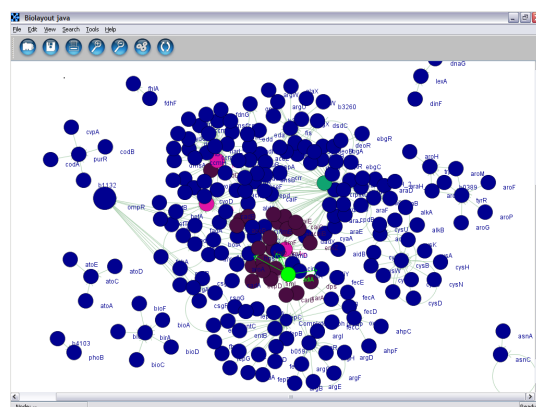
Figure 3.7. Common to visualisation employing node-link graphs, *ATV Viewer*, developed by [186], exhibits severe occlusion in parts of the graph on the right while other areas lie empty. The smaller graph on the left makes better use of visual cues, allowing more intuitive analysis to be performed. (Images printed with permission using sample data files and the *ATV Viewer* applet at: <http://www.genetics.wustl.edu/eddy/atv> (last viewed Jul 2006).)

BioLayout JAVA⁹, developed by [65], uses directed network graphs to visualise relationships in biological data, with functionality provided for interactive modification of layout. *Classes* may be created to describe different data types; allowing colour coding to be used for data classification. Edges containing relationships between element pairs may also be weighted and/or colour coded based on strength of relationships occurring. The visualisation system may direct searching for additional information on a node to the default web

browser on a user's machine, using the name of the node as the search term in a user-specifiable data source. Finally, changes made to a graph can be saved in a reloadable file. Figure 3.8 illustrates the results of interactive modification of a graph plotted for metabolic pathways in *E. coli*.



(a) Initial layout of biological data using a connected network graph in *BioLayout JAVA*



(b) Result of interactive modification of the graph in figure 3.8(a). Zoom is also used to reduce empty space and provide greater magnification to ROIs.

Figure 3.8. *BioLayout JAVA* is used to visualise metabolic pathways in *E. coli*. Nodes assigned to three user-defined classes are colour-coded. A sub-string search highlights 2 nodes in the graph and (directed) links from search hits to related data.

(The sample data files and the *BioLayout* application can be obtained from the *BioLayout* web site at: <http://cgg.ebi.ac.uk/services/bioblayout>

3.3 Limitations in tree graph visualisation

A major limitation in tree graph visualisation is poor scalability: increasing occlusion with data set size significantly reduces the usefulness of overviews [69, 159]. 2D graphs are generally only able to display a few hundred nodes before occlusion renders overviews unusable. Hierarchical, node-link graphs do not make optimal use of screen space; the area surrounding the root of a tree lies empty while density of nodes increases toward the leaves [139]. Occlusion when it occurs results in data objects being hidden behind others, and crossing of links obscures relationships between elements.

A number of solutions are available for combating occlusion in tree graphs. Pan and zoom provide simple solutions to occlusion in ROIs, but result in a loss of the context provided by the overview. Using coupled windows to provide an overview while detail is examined in the main window helps to regain context. However this requires extra cognitive effort mapping between the two visualisations. Data and dimensionality reduction suppress less important data and/or attributes, or clusters like data into composite nodes, reducing the number of objects drawn to the screen.

Another solution is to draw visualisations using the larger amount of space available in 3D. Distant elements in 3D space are however obscured by objects closer to the viewpoint; as in 2D, this may lead to misinterpretation of data content [91], resulting in inaccurate

mental models being formed of data structure. Further, zooming in to sub-structures in a visualisation to analyse ROIs in detail sacrifices the context of the overview.

Walrus, developed by [106], may present a solution to space limitations in 2D, making use of the exponential increase in space provided by the hyperbolic projection in 3D to draw trees containing hundreds of thousands of nodes. [152] describe the *Cone Trees* system, which also takes advantage of the larger amount of space in 3D to display large data sets. Data structures in 3D can be rotated to move to elements occluded by other objects closer to the viewpoint, an option which does not exist in 2D. Links that cross paths in 2D may be placed in different, non-intersecting planes, so that their paths remain distinct in 3D.

3.4 Summary

This chapter continued the review of information visualisation systems, with a focus on graph visualisation tools. A sample of tools and techniques developed to provide analysis of large amounts of inter-related data was presented, comparing techniques used in different applications to provide solutions to problems encountered in data analysis.

Chapter 4 looks at biological ontologies and tools developed to aid analysis of bioinformatics data.

Chapter 4

Bioinformatics data analysis

The large amount of data generated from experiments includes associated information such as annotation of results, experimental conditions, equipment used, identities of researchers, and the results of processes performed on data obtained during experimentation. Challenges in bioinformatics data management are further complicated by the sometimes seemingly conflicting expert annotation based on domain knowledge, used to describe experiment results and other data. Further, new biological data is received on a constant basis, so that changes to existing knowledge occur with time; new experiments may invalidate previous theories and hypotheses formulated [124, 149]. Support is required for timely update of data stores so that such changes are identified and further analysis performed, modifying theories and hypotheses as required. Poor integration between data sources however makes it difficult to perform seamless data updates and ensure consistency between data sets [9, 29]. Different terms used and/or different interpretation of terminology used to label and annotate data further complicate integration of data from multiple sources.

4.1 Ontologies in bioinformatics

Bioinformatics research involves large amounts of heterogeneous data with varying levels of accuracy and stored using different formats. Being a multi-disciplinary field, bioinformatics data is further compounded by a significant number of methods and terms used to label and/or annotate data, and consequently, in interpreting information [84]. This poses difficulty in cross-referencing and data integration [90, 167]; effective research requires data analysis tools that are able to communicate with each other, and read and process data from multiple sources [19]. Data reuse and persistence increase where a common language is *spoken*; access to external data sources enriches existing data [124, 185], aiding the formulation of new theories and/or validating existing ones [90]. Redundancy in information may be removed, and conflicts resolved more easily.

Ontologies, which may serve as data dictionaries or controlled vocabularies, store semantic information about terms particular to a knowledge domain [74] and the relation-

ships between them. Ontologies provide a framework for expression of abstract ideas or concepts, serving as a platform for common (understanding of) knowledge within a community. The semantic links formed between different data elements aid navigation through and exploration of data [84].

Biological ontologies may serve as digital representations of organisms [13]; such structured vocabularies provide a standard reference framework that eases comparison of multiple data sets [87, 105]. A sample of well-known biological ontologies, including the Gene Ontology and the Foundational Model of Anatomy, are described in § 4.2.1.

Ontologies may be fairly general, making them more reusable and more widely applicable, referred to as upper-level ontologies, two well-known examples of which are *Cyc*¹ and the *Dublin Core*². Alternatively, specialised or domain-specific ontologies limit general use but provide richer semantic information for a narrow field.

Upper-level ontologies, which define concepts at an abstract level, are useful for providing mappings between more specialised ontologies belonging to related fields, or for serving as a starting point for creating domain-specific ontologies. Using standardised ontologies for data annotation helps to differentiate similar terms used to refer to dissimilar data, and conversely, dissimilar terms used to describe identical or closely related data. The common language provided by ontologies eases data exchange and analysis, especially in automated systems [90], which do not have the support of humans who may bring domain knowledge to bear in clarifying ambiguities in data. [14], however, recognise that reuse of ontologies is very low, not only because a large proportion of existing ontologies are fairly domain-specific, but because of inherent complexity and the lack of integration among the tools that provide access to these ontologies, and among the ontologies themselves.

Using the same ontology to compare different data sets will highlight similarities and differences between them. On the other hand, analysing a single data set using multiple ontologies reveals alternative perspectives, highlighting different relationships occurring within the data. Different types of relationships may be defined between elements pairs in an ontology, common examples being *is-a* and *part-of*. Ontologies that describe part-of relationships between elements, for example, provide a natural method for data classification based on a hierarchical structure, where sub-components form part-of relationships with their parents. For completeness, all sub-components of a specified (parent or super) component should together make up the whole component. It should be noted that alternative methods for data classification may mean that a single component could be defined such that it forms a part of more than one named structure. This results in a hierarchical structure with directed links, allowing multiple paths to be followed from any such component to the root

¹More information on the Cyc knowledge server can be found at: <http://www.cyc.com/cyc/technology/whatis-cyc>.

(Note that this and all other web addresses referred to in this chapter were last viewed in July 2006.)

²More information on the Dublin Core Metadata Initiative can be found at: <http://www.dublincore.org/about>

of the hierarchy. Graphical representations of such data sets may be used as navigation aids in data exploration, revealing the structure of the ontologies thus formed and the different relationships they contain. § 5.3.2 and § 5.3.4 contain a discussion on the use of visualisation to present alternative structural representations for ontologies.)

4.1.1 Anatomy Ontologies

[85, 156] make a case for the use of the anatomies of individual organisms as a base for data exchange and integration in biological research; because anatomy is a pillar of biology using anatomy data to create ontologies should aid analysis of genomic and other biological data for different organisms. This argument is supported by the quest for standardised anatomy ontologies in biological, medical and pharmaceutical research [18], where differences in fields of application result in even greater variation in accuracy and detail during collection, annotation and storage of data [33, 94], even where the same anatomical components are being referenced [136]. Automated search and query cannot be guaranteed to retrieve information required; it is necessary to make use of domain knowledge to verify data analysis, decreasing efficiency (and increasing the probability of error). Data exchange and retrieval also become more difficult and time consuming, as they cannot be fully automated.

Mapping the physical components that make up the cells, tissues and organs in an organism to standard ontologies eliminates or at least greatly reduces the problem of inconsistency, and aids IR and data exchange, leading to more effective analysis. Anatomy ontologies store terms that may be used for annotation of gene expression data, based on anatomical components within which genes are expressed. Other biological concepts may also be defined with reference to the anatomical components they are associated with, aiding cross-referencing and analysis involving multiple data sources.

[94] define an anatomy ontology as “*a structured vocabulary of anatomical entities in which the terms have unique identities and relate to each other in meaningful ways*”. The aim here is to resolve inconsistency in the description of similar or identical concepts in biological data, and decrease difficulty searching for information from different data sources [14], especially for automated analysis.

A major aim in building ontologies and structured vocabularies is to ease data exchange and integration. However because ontologies are often independently created and do not always make reference to those already in existence, they solve at best a part of the bigger problem. Furthermore, methods used to verify terms, often based on research domains and expertise, naturally vary between projects. One solution currently in use to increase integration is to provide mappings between elements that refer to related or identical terms in different ontologies [156].

Another attempt to resolve this issue is the public-access Standards and Ontologies for

Functional Genomics (SOFG) group's Anatomy Entry List (SAEL) [136]³. The SAEL aims to identify between 100 and 150 *core* elements, to form a controlled vocabulary that can be used directly for annotation of gene expression data at a low level of detail, or point to other more detailed ontologies as required. The initiative aims to improve integration and/or cross-referencing of the large number of independently created ontologies and structured vocabularies in existence, each of which is looking to solve what is a common problem in data retrieval and analysis.

4.2 Tools and techniques for bioinformatics data analysis

Research fields working together in *Bioinformatics* can be broken up into three main areas: *Biology*, *Computer Science* and other sciences. Different researchers will have different information requirements [84], based on current field of work and domain knowledge or expertise; it should be ensured that software developed for analysis of bioinformatics data is usable by the researchers with varying backgrounds performing analysis from different perspectives. It is important also to remember that there may be a significant number of users without a background in CS and who may have limited skills for working with complex computational data analysis and visualisation tools [13, 36, 45]. Further, most tools are only used occasionally, for the specific aspect of analysis they are best suited for, so that most users remain novices or only casual users for a large number of tools. Varying data formats for both input and output and differences in accuracy, among others, further increase difficulty transferring data and learning between systems [38].

It is important to develop simple, intuitive interfaces that use standardised or at least similar methods for data input, processing and output, so that time and resources required to set up and learn to use new tools are kept to a minimum [165]. This is even more important for the use of online applications where short learning curves and low reliance on external support are expected. Access to and use of non-standard systems, data formats and sources should be transparent to users; the aim of bioinformatics is to provide a service that harnesses technology to improve research in biology.

A large number of computational tools have been developed to aid biological data analysis, starting from the first Perl-driven programs working off Linux boxes [167] to current sophisticated tools written in a host of languages and that employ complex algorithms, providing a range of interfaces, from simple command-line interfaces (CLIs) or forms-based front ends to high-end, customisable GUIs. Applications that provide advanced imaging and visualisation for analysis of the complex, multi-dimensional data involved in bioinformatics are also being developed.

Current bioinformatics tools largely provide simple graphical or forms-based interfaces to databases, that also serve as pipes between different data sources and/or applications.

³See also the SOFG web site at: <http://www.sofg.org>

Support is provided for data management, processing and intuitive analysis that highlights relationships within data and facilitates the extraction of new information from data, to aid the formulation of and/or confirmation or refinement of research hypotheses [2, 13].

A second significant development area looks at structural and functional analysis of genome data, developing tools for performing gene sequence alignment and similarity and homology mapping. The following sections describe a sample of tools developed for bioinformatics data analysis.

BLAST

The *Basic Local Alignment Search Tool* (BLAST) is used to compare newly discovered gene and protein sequences with existing ones. Mapping areas of similarity, based on evolution, aids the identification of gene structure and function [9]. A large number of implementations of BLAST are available online, most of which use forms interfaces for data input.

Clustal

Clustal programs are used to perform alignment of multiple protein sequences. Phylogenetic trees are often used to visualise evolutionary relationships discovered within the data. A number of implementations of *Clustal* are available online⁴.

Ensembl Genome Browser

*Ensembl*⁵ is a joint project run by the European Bioinformatics Institute at the European Molecular Biology Laboratory⁶ (EMBL-EBI) and the Wellcome Trust Sanger Institute⁷. *Ensembl* comprises a set of public-access tools used to manage annotation for the genomes it stores, providing support for data mining, structural and functional analysis of genes and proteins, and gene sequence alignment and similarity searching.

BioMart is a data management and mining tool that forms part of the *Ensembl* suite, with support for complex querying. Online use is available, employing a form embedded in a web page, or offline use from a textual or graphical interface on a standalone version developed using Java and Perl.

Phylogeny tree drawing tools

Dendrograms and cladograms are commonly used to generate the phylogenetic trees useful for describing evolutionary relationships within biological data.

⁴Clustal sites include EMBL-EBI: <http://www.ebi.ac.uk/clustalw> and the Institut Pasteur: <http://bioweb.pasteur.fr/seqanal/interfaces/clustalw.html>

⁵The Ensembl Genome Browser can be found at: <http://www.ensembl.org>

⁶The EBI web site can be found at: <http://www.ebi.ac.uk>

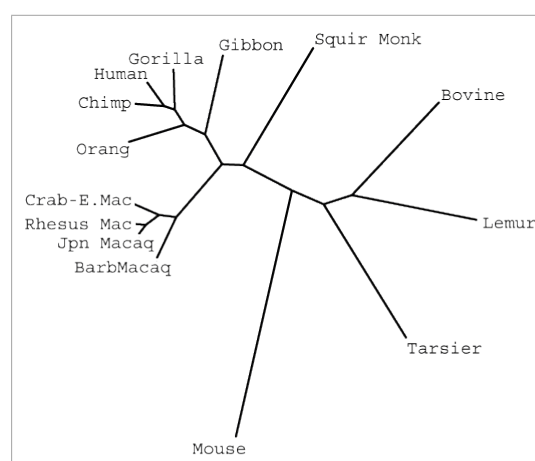
⁷The Sanger Institute web site can be found at: <http://www.sanger.ac.uk>

Phylodendron⁸ is an interactive drawing tool built using Java, for drawing phylogenetic trees, available both as a standalone package and for use from a web server. Figure 2.13 shows a tree diagram drawn using the web service on the *Phylodendron* web site.

PHYLIP⁹ is a collection of menu-based programs developed in C to aid the identification of evolutionary relationships within data. Output may be visualised using the drawing programs provided for generating phylogenetic trees, (an example of which is shown in figure 4.1), some of which include options for interactive construction and modification of tree layout.

A compendium of phylogeny tools can be found on the web site of the Felsenstein laboratory at the University of Washington, Seattle, at: <http://evolution.genetics.washington.edu/phylip/software.html>.

Figure 4.1. A phylogenetic tree drawn using the *DrawTree* application that forms part of the *PHYLIP* suite, to show evolutionary relationships between 14 mammals. (Image obtained from and used with permission of Joe Felsenstein, Departments of Genome Sciences and Biology at the University of Washington.)



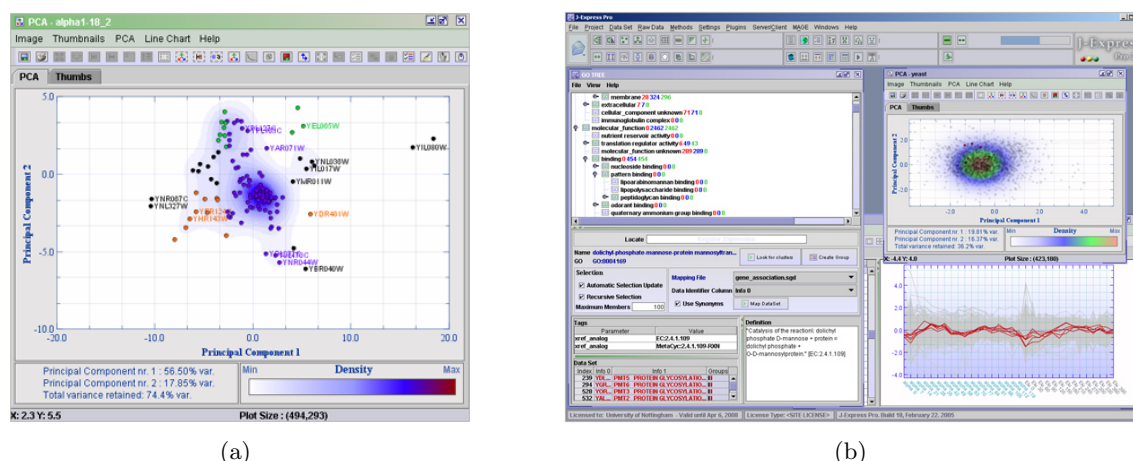
Space Explorer

Space Explorer was developed to combat the limitations of dendrograms for visualisation of (large amounts of) gene expression and associated biological data. *Space Explorer* [89] makes use of PCA to perform multi-dimensional scaling, visualising the results using a spring layout. Clustering of like data is used with colour coding to show similarity in data attributes, simultaneously highlighting outliers or anomalies. [89] found that laying out the gene expression data they studied using coordinates in Euclidean space provided a better indication of similarity within data than was obtained performing hierarchical clustering using dendrograms.

Space Explorer uses VRML to describe the 3D world used to display the multi-variate gene expression data studied. Shape and texture of objects drawn provide additional properties for encoding data. Using VRML allows analysis within web browsers, and also harnesses the visual and navigational cues available in VR worlds. Landmarks and viewpoints

⁸See Don Gilbert's page at the Department of Biology, University of Indiana: <http://iubio.bio.indiana.edu/soft/molbio/java/apps/trees>

⁹See the PHYLIP home page at the Departments of Genome Sciences and Biology, University of Washington, Seattle: <http://evolution.genetics.washington.edu/phylip.html>



¹¹The GO Consortium can be found at: <http://www.geneontology.org>

online. The following sections look at some of the most widely referenced ontology databases and a sample of tools dedicated to management of ontologies.

The Gene Ontology

The Gene Ontology¹¹ (GO) is a public-access data source that stores information on genes and gene products using a set of controlled vocabularies that describe gene function and structure, and biological processes associated with gene products. GO was developed to address difficulty referencing information from different data sources, due to underlying differences in terminology used for annotating data. A common language is provided that aids inter-referencing and integration of multiple data sources [9], using unique identifiers for each element described.

Tools are provided for use with GO, for creation and management of ontologies and for associating terms within GO with those found in other databases [93]. Online and standalone browsers developed to support data upload and querying of the GO database are described below.

AmiGO¹² is a Java-based ontology browser developed by the *Berkeley Drosophila Genome Project*. *AmiGO* uses a forms interface for online searching and browsing of GO terms and relationships between terms. Information retrieved may be viewed using a collapsible, hierarchical text index or a node-link graph, as shown in figure 4.3.

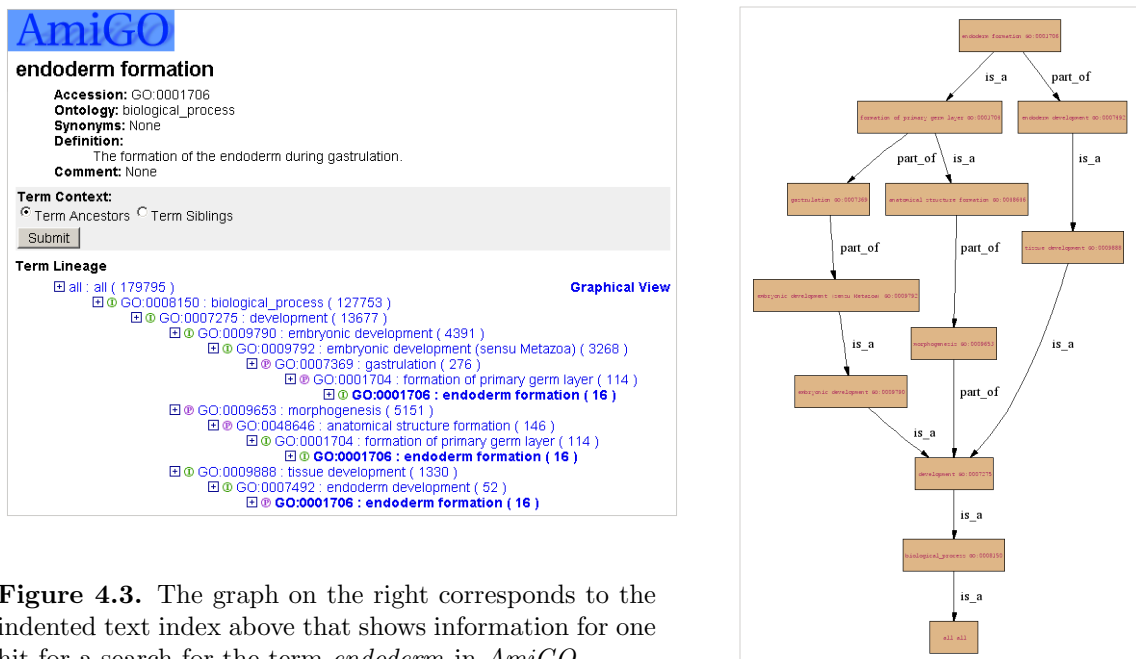


Figure 4.3. The graph on the right corresponds to the indented text index above that shows information for one hit for a search for the term *endoderm* in *AmiGO*.

¹²The Gene Ontology Software and Databases and the Berkeley Drosophila Genome Project at: <http://www.godatabase.org/dev>

COBrA¹³ is a Java-based anatomy ontology browser, developed as part of the XSPAN project, for browsing, analysing and editing GO and OBO (Open Biomedical Ontologies) ontologies. *COBrA* was developed to provide support for creating mappings between equivalent concepts in independent ontologies. Additional functionality translates terms in GO and OBO to Semantic Web¹⁴ languages such as the Web Ontology Language, OWL. *COBrA* employs a collapsible, hierarchical index coupled with a text search field for browsing data and for IR.

QuickGO¹⁵ is a forms-based, online browser developed at the EBI for querying GO. The information retrieved is used for annotating data in the Universal Protein Resource (UniProt) and Ensembl databases, as part of the GO Annotation¹⁶ (GOA) project.

The MGI GO Browser¹⁷ uses a forms interface to upload data to GO or retrieve GO terms for annotation of gene expression data for the laboratory mouse, as part of the Mouse Genome Informatics (MGI) project at the Jackson Laboratory.

DAG-Edit¹⁸ was developed for the management of GO ontologies. Plug-ins provided for use with *DAG-Edit* include a graph viewer based on the *Graphviz*¹⁹ library developed at AT&T Research, to visualise ontologies using DAGs. *OBO-Edit*²⁰, is being developed to replace *DAG-Edit*, and is currently at the beta stage of development.

Open Biomedical Ontologies

OBO²¹, with links to GO, provides a compendium of biological and medical ontologies. The OBO site provides links to related projects and other resources for searching and analysing ontology data.

The Foundational Model of Anatomy

The Foundational Model of Anatomy²² (FMA) provides a knowledge base containing over 100,000 terms and more than 2 million relationships between components, that describes the structure of the human anatomy. The FMA provides a framework that serves as a reference point for bioinformatics and biomedical research [85]; to aid integration between multiple data sources and views, and standardisation of terminology in use. Although

¹³See COBrA - An Ontology Browser for Anatomy at: <http://www.xspan.org/cobra>

¹⁴See the W3C Semantic Web pages at: <http://www.w3.org/2001/sw>

¹⁵The QuickGO GO Browser can be found at: <http://www.ebi.ac.uk/ego>

¹⁶See GOA @EBI at: <http://www.ebi.ac.uk/GOA>

¹⁷The Mouse Genome Informatics site can be found at: <http://www.informatics.jax.org>. Note that EMAP has links to the Jackson MGI project.

¹⁸See <http://www.geneontology.org/GO.sourceforge.links.shtml#dag>

¹⁹The Graphviz home page can be found at: <http://www.graphviz.org>

²⁰See <http://www.geneontology.org/GO.sourceforge.links.shtml#obo>

²¹See the Open Biomedical Ontologies web site at: <http://obo.sourceforge.net>

²²See the FMA web site at: <http://sig.biostr.washington.edu/projects/fm>

machine-based to enable automated searching and referencing, the FMA is presented such that it is easily read and analysed by humans, using text IDs and descriptions to provide a symbolic representation of the information it contains. Data is stored in a relational database managed using the *Protégé* 3.0²³ ontology editor. [156] provide a detailed description of the FMA.

The OpenGALEN Ontology of Human Anatomy

The OpenGALEN Ontology of Human Anatomy²⁴ was built to provide a machine-based source of information that promotes integration of the multiple data sources and applications available for data analysis in clinical medicine. GALEN (the Generalised Architecture for Languages, Encyclopaedias and Nomenclatures in Medicine), makes use of GRAIL (GALEN Representation and Integration Language) to present knowledge in clinical medicine in the GALEN Common Reference Model (CRM). This model recognises the importance of a formal, controlled language that reduces misinterpretation of data while allowing information to be presented from different perspectives. [146] describe the design of the GALEN ontology.

TAMBIS

The Transparent Access to Multiple Bioinformatics Information Sources Project²⁵ (TAMBIS) is included here for completeness — *TAMBIS* is no longer supported and the software is not available for use. *TAMBIS* was developed to provide a central online resource for transparent, visual querying of biological and bioinformatics data, using the *TAMBIS Ontology* (TaO) to aid users in phrasing effective queries [90]. TaO was also used to parse information and determine similarity in data retrieved from multiple sources. Applications such as *TAMBIS* remove the burden of data sourcing and management from researchers, hiding effort required to integrate the myriad data sources based on varying underlying schemas and with differences in data annotation. Even though ability to formulate complex queries using formal query syntax is still an advantage, support for natural language querying and reprocessing of queries is a significant benefit to especially non-technical researchers.

Protégé

*Protégé*²³, described in [135], is a leading Java-based application incorporating a set of tools for managing ontologies and knowledge bases. *Protégé* has a wide user base, is supported on a number of platforms, and is customisable and extensible. *Protégé* provides forms and an alternative graph-based interface for manipulating ontologies. The latter has the advantage of providing a visual representation of the relationships between elements, as illustrated in figure 4.4.

²³The Protégé Ontology Editor web site can be found at: <http://protege.stanford.edu>

²⁴See the OpenGALEN web site at: <http://www.opengalen.org/open/crm/crm-anatomy.html>

²⁵The repository for the TAMBIS Project can be found at: <http://imgproj.cs.man.ac.uk/tambis>

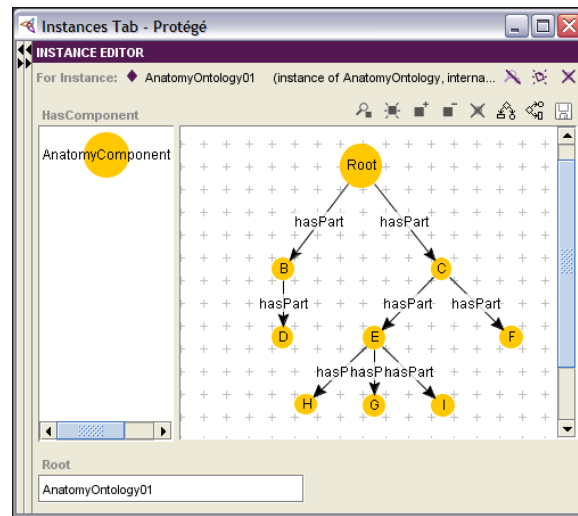


Figure 4.4. The *Graph Widget* plug-in for the *Protégé* ontology editor is used to draw a simple tree to show *part-of* relationships between nodes in a pseudo anatomy ontology. Element attributes and relationships between elements are encoded using colour, shape and size. (Snapshot printed with permission from a working version of *Protégé* 3.1.1).

A number of visualisation plug-ins have also been developed to aid data analysis in *Protégé*, some of which are described in the following sub-sections.

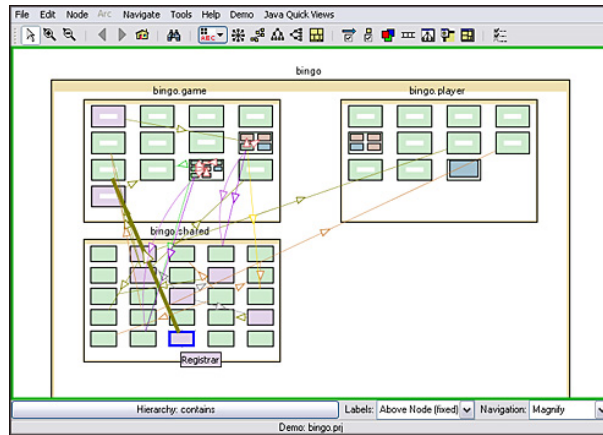
Jambalaya²⁶, which uses graph visualisation to reduce cognitive load during interaction with ontologies [66], is a tool created by integrating *Protégé* with *SHriMP*, the Simple Hierarchical Multi-Perspective visualisation technique. Data overviews are generated that provide users with context during navigation, to manage the occlusion and disorientation that occurs especially in exploration of large data sets. An advantage in *Jambalaya* is design for customisability and extensibility. *SHriMP*, illustrated in figure 4.5, was designed to support exploration of information spaces, and makes use of *Piccolo*'s continuous zoom library [22], to provide a modified F+C layout for nested tree graphs.

SHriMP was developed using Java, and *Piccolo*'s continuous zoom library includes a Java implementation. *SHriMP* obtains data abstraction by collapsing sub-trees into composite nodes, reducing occlusion in graphs generated. Pan and (geometric) zoom are provided for navigation to and within ROIs, and hyperbolic and semantic zoom to allow users to retain context during data exploration. Recording user paths through data provides history sessions that allow users to move back to previous views. Colour, shape and size of nodes are used to encode data attributes. [171] provide a detailed description of the features of *SHriMP* available in *Jambalaya*.

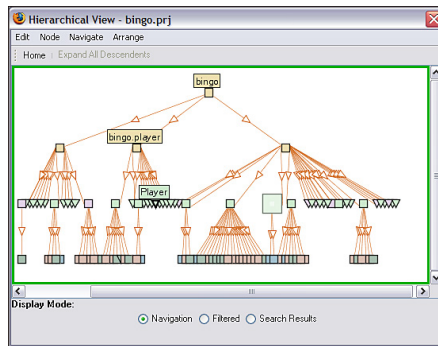
TGVizTab makes use of the *TouchGraph*²⁷ library, developed in Java, to generate interactive network graphs that use spring layouts to cluster like data [4]. Colour is used to

²⁶More information can be found on Jambalaya and SHriMP from the CHISEL group's web site at: <http://www.thechiselgroup.org>

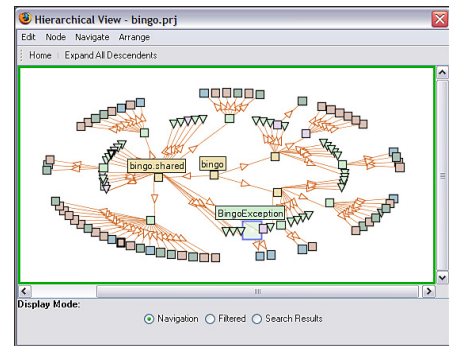
²⁷See the TouchGraph home page at <http://www.touchgraph.com>



(a) The default nested data view in *SHrIMP* used to visualise a software package.



(b) Vertical layout of the node-link graph showing relationships between classes and instances in the package in figure 4.5(a)



(c) Radial equivalent for the graph in figure 4.5(b)

Figure 4.5. A demonstration of some of the functionality provided in *SHrIMP*, which is incorporated into *Protégé* to form *Jambalaya*.

(Snapshots printed with permission from a demonstration version of *SHrIMP* downloaded from the CHISEL web site at: <http://www.thechiselgroup.org/shrimp>)

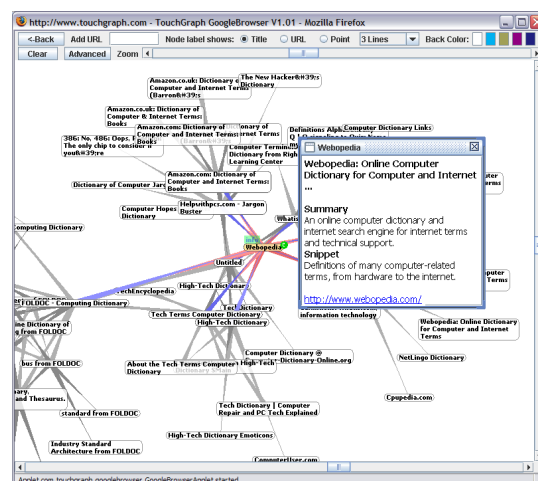
encode data, and geometric zoom is available for detailed analysis of ROIs. Complexity of graphs can be controlled by successively collapsing children into parent nodes. Functionality is also provided for saving graphs and system state, for continuous and collaborative analysis.

Figure 4.6 demonstrates the use of the *TouchGraph* library to visualise relationships between documents linked to and within the *FOLDOC* reference pages.

OntoViz uses the *Graphviz*¹⁹ library to visualise structured data using directed graphs and networks. *OntoViz* makes use of the languages *dot*, *lefty* and *neato* to draw graphs. Colour and shape of nodes is used to encode data attributes, and abstraction is obtained by clustering like data together. Graphs may be drawn in 3D by making use of functionality developed to write data structure to VRML files.

Figure 4.6. *TouchGraph* is used to lay out documents within and linked to the FOLDOC web site using a network graph. Selecting *Webopedia* highlights colour-coded (hyper)links from the node to other sites, and clicking on the *info* button attached to a node of interest brings up detail on the selected document in the coupled sub-window shown.

(Snapshot printed with permission from the *TouchGraph GoogleBrowser* at: <http://www.touchgraph.com>)



OntoRama

[64] describe the Java-based tool, *OntoRama*, shown in figure 4.7, that uses a node-link hyperbolic layout to browse ontologies, allowing the display of several times more data nodes in the display than would be possible for the equivalent Cartesian layout. Multiple inheritance is visualised by *cloning* nodes, to reduce the crossing of links that might otherwise occur; this results in a hierarchical structure even for data with a large amount of inter-linking. *OntoRama* can also draw multiple trees simultaneously, to form a *forest* for related but fairly disconnected data. A corresponding collapsible, hierarchical index provides an alternative to visualisation using the hyperbolic layout. Users may zoom in to either data representation to analyse ROIs in more detail. Finally, forms-based querying is augmented with the ability to click directly on nodes in the graph to generate queries.

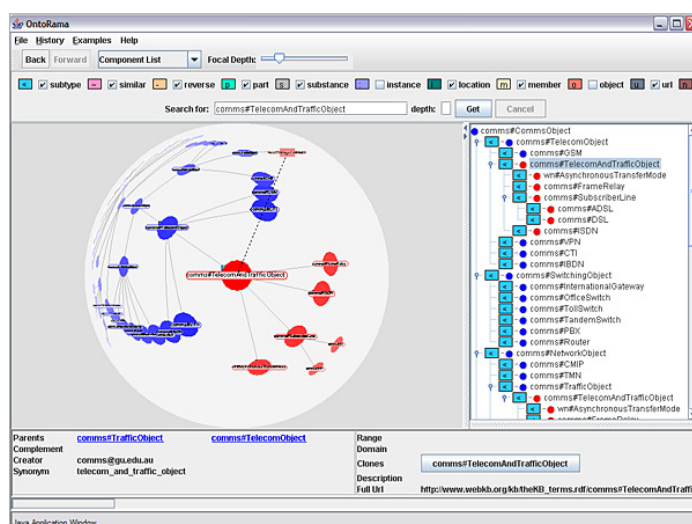


Figure 4.7. A demonstration version of *OntoRama* visualises the ontology for a communication system using a hyperbolic layout. Cloned nodes are encoded in red; clicking on the focus draws a broken line to its clone. Textual detail for the node of interest is printed to the lower section of the window, and the corresponding entry is highlighted in the collapsible index.

(Snapshot printed with permission from a demonstration version of *OntoRama*, using a sample data set downloaded from the original Ontorاما web site.)

4.3 Applications of bioinformatics

One of the most important contributions of bioinformatics to science is the support it provides for data management, dissemination and analysis. The following sections look at the contribution of bioinformatics to fields in or related to biology.

Genomics

Bioinformatics provides data storage, management and analysis of the large amounts of complex data generated in genome research. Previous sections have looked at research into development of intuitive data analysis tools for effective IR. Tools for structural and functional analysis of gene expression data and alignment of gene sequences are constantly being developed to retrieve knowledge stored in newly uncovered information.

Molecular biology and other life sciences

Bioinformatics tools are used to mine the large amounts of data generated in biology and other related subjects, to retrieve the knowledge stored within the data. Bioinformatics improves research in structural and molecular biology, by providing advanced computational and imaging tools that simulate different biological structures.

The study of evolution contributes to knowledge about related organisms and their development. Neural networks used in artificial intelligence (AI), modelled on biological neurons, are used in machine learning, and genetic algorithms are used to uncover solutions for optimisation problems.

Medicine and pharmaceuticals

An organism's genetic makeup defines its physiology and affects its development. Genes and gene expression data are being studied to determine susceptibility of specific genes or entire genetic make-up to disease [111], and level of resistance to drugs [131]. This may lead to the development of preventative medicine and cures tailored to the specific needs of individuals, based on analysis of their genetic make-up. Drugs may be developed to target specific genes, resulting in more effective treatment of disease [105, 121].

Bioinformatics improves the analysis of complex biological data, helping to retrieve information that may be applied directly to research in the pharmaceutical industry. The ability to map structure and function of gene expression data across organisms, using previously identified structures, results in more efficient research and experimentation. Curative properties in elements are more quickly identified, and targeted experiments can be performed that provide more effective drugs and treatment for disease and other physiological conditions.

Researchers are increasingly relying on secondary data sources to confirm hypotheses formulated, or to re-evaluate prior conclusions drawn. The use of ontologies aids the research

required for development of new drugs; associating semantic content with the diverse terms used to describe information aids understanding and eases retrieval of (new) knowledge stored in data obtained from experimentation and other research activities [18].

Biotechnology and bioengineering

Research into micro-organisms is useful in industry, in the manufacture of food reliant on bacteria for fermentation, and for managing waste. Research is also looking at the generation of alternative sources of energy based on processes occurring in micro-organisms.

Agriculture

Advanced research is used to improve yield and resistance of crops and farm animals to pests and disease.

4.4 Ethical issues in bioinformatics research

There are a large number of social and ethical issues involved in genetics research. Sourcing and treatment of physical specimens raise questions about the right to experiment on especially embryos. The use of human specimens raises even more questions, especially where subjects are used without their knowledge or consent [92].

Debate continues about potential risks from eating genetically modified crops and animals. Gene therapy, which may provide cures for or prevent conditions that currently have limited effective treatment, is also a source of controversy, especially with the potential for misuse and difficulty predicting long-term effects on subjects.

How information is obtained and used is also important. The ability to map the genetic make-up of an individual and thus determine their susceptibility to disease and other physiological conditions may lead to improved treatment of such conditions. However availability of this information to insurance companies and industry may lead to discrimination and stigmatisation based on perceived ability to perform a role or susceptibility to various medical and physiological conditions [7].

More information on ethical issues in bioinformatics is available from the Ethical, Legal, and Social Implications (ELSI) Program at the United States National Human Genome Research Institute web pages²⁸.

4.5 Summary

This chapter examined the role anatomy ontologies play in biological research: aiding the integration of multiple data sources and the retrieval of knowledge stored in data. A sample

²⁸The ELSI web pages can be found at: <http://www.genome.gov/PolicyEthics> and the Policy and Ethics pages at: <http://www.genome.gov/10001618>

of bioinformatics tools were examined, looking especially at the management of biological ontologies.

The chapter concluded with a brief look at the fields in which results of research in bioinformatics are applied and the ethical and social issues associated with bioinformatics research.

The remaining chapters examine the problem domain this thesis explores, detailing the development and evaluation of a visualisation solution for the information requirements identified.

Chapter 5

Challenges in the analysis of anatomy ontologies

Chapter 4 gives an introduction to the challenges faced in research that requires cross-referencing and integration of multiple, independently created data sets, and the use of ontologies to help resolve these challenges. Cross-referencing ontologies however presents a problem of its own: even though they aim to provide common reference frameworks describing data in knowledge domains, because ontologies are also often independently created, especially where they cater to a narrow field of use, ontologies may provide only a limited solution to the data integration required for effective research.

One requirement in the use of anatomy ontologies is the ability to map anatomical components to the varying concepts and terminology employed in different biological data sets. Knowledge and experience of domain experts play a significant role in determining similarity within data, supplementing lexical analysis. Another requirement is the ability to trace continuous or temporal relationships in data, across multiple data sets; research and analysis in biology often require the use of data collected over periods of time.

Identifying relationships within data becomes more difficult as the amount of data being analysed increases, largely due to difficulty obtaining a useful overview of data structure. Where data is presented in textual format high cognitive (memory) load is associated with analysis that attempts to obtain an understanding of the relationships that occur within the data. Multiple relationships between data elements increase complexity of data structure; analysis requires understanding of data structure and the existence and types of relationships occurring within the data. This thesis looks at harnessing visualisation to provide intuitive solutions to the analysis required, using advanced human perception to build an understanding of data structure and aid the identification of relationships within individual and that cross multiple data sets.

EMAP and XSPAN are research projects that make use of anatomy ontologies to perform genomic research. Data analysis and information requirements for the two projects include those typical to similar research in bioinformatics: EMAP and XSPAN involve the

identification of the different types of relationships that occur between anatomical components, to aid understanding of the information contained within each ontology. Knowledge obtained is mapped to analysis of related data, to be used, for example, in the inference of structure and function of anatomical components in one organism based on similarity to genes expressed in another related organism.

The EMAP and XSPAN projects provide sample anatomy ontologies and access to researchers in genomics, to aid the development of intuitive, usable options for analysis, working with typical users of the ontologies of interest. The following sections provide an introduction to the two projects and detail their specific information requirements.

5.1 The Edinburgh Mouse Atlas Project

The mouse is a model organism for research on the genetic make-up of mammals [28, 29]. Data stored on the mouse genome, however, typical to biological data, is heterogeneous and exists in large amounts. This includes data not directly relevant to analysis, such as details of experimental conditions. Further, the data from different sources is stored using a variety of formats and may use different *language*, increasing difficulty in data exchange and analysis [54].

The Mouse Atlas Database has been developed to serve as a public-access framework for studying gene expression data for the mouse [54]. The EMAP data comprises the 26 Theiler Stages of development of the mouse embryo, stored in ontologies and displayed using hierarchical text indices [54] that describe *part-of* relationships between components. (Stages 27 and 28, with information on newborn and post-natal development, in addition to information on other stages of development, are stored in the Gene Expression Database (GXD) at the Jackson Laboratory¹). EMAP also stores reconstructed 3D models of the mouse embryo mapped to the ontologies describing the different components that make up the mouse anatomy at each stage of development [15]. Mapping gene expression data as it is uncovered to corresponding components in the anatomy ontologies provides intuitive representation of the spatio-temporal data the ontologies describe [17, 13]. The ontologies in turn serve as annotation for the image data obtained from the models of the embryos, allowing text searching to be performed where advanced imaging and pattern recognition are not available, or in addition to results from image analysis. The snapshot in figure 5.1 shows one of the tools currently provided for analysis of the EMAP data.

The EMAP section browser comprises three main sections, including a collapsible, indented text index that lists components in the anatomy ontology for each stage of development of the mouse embryo. Simple text searching is available, with closest hit found highlighted in the index and mapped to corresponding regions on 2D slices cut out of the

¹Information on the GXD can be found at: <http://www.informatics.jax.org/mgihome/GXD/aboutGXD.shtml>

(Note that this and all other web addresses referred to in this chapter were last viewed in July 2006.)

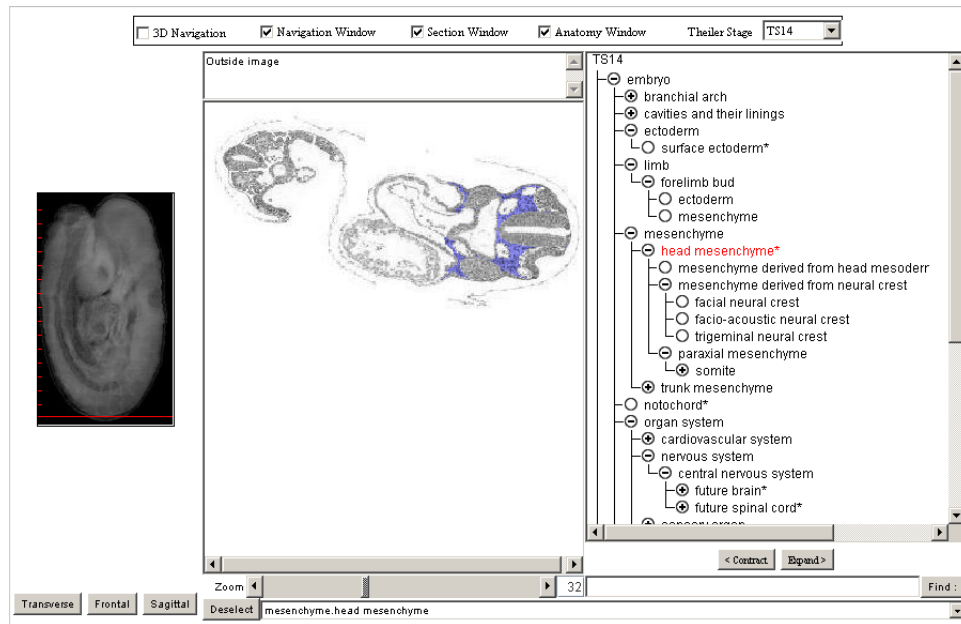


Figure 5.1. A snapshot of the EMAP browser shows a 2D slice cut out of the 3D model (along the red line) of the mouse embryo for TS14. Selecting the entry for the *head mesenchyme* in the text index highlights the region that maps to the component on the 2D slice in blue. The text field at the bottom of the browser allows simple text searching within the current ontology.

reconstructed 3D models of the embryos, the latter forming the other two sections of the browser. Alternatively graphical querying may be performed: clicking directly on the image to select an ROI on the 2D slice highlights the component it maps to in the text index. Information on genes expressed in anatomical components of interest may be retrieved by choosing the option to extend searching to the Edinburgh Mouse Atlas Gene Expression (EMAGE) database and the GXD.

The browser shown in figure 5.1 may be used directly from the EMAP online repository², and a second browser with more advanced functionality for searching within EMAP or local databases, shown in figure 5.2, may also be downloaded from the same location. Additional resources for data analysis can also be found on the EMAP site³.

5.1.1 Structure of EMAP anatomy ontology

The ontology for each developmental stage defines all anatomical components that are visible when the stage is entered and those that develop during the time span the stage covers. The EMAP anatomy ontology is structured based on *part-of* relationships between nodes [15]: all *child* or *sub-components* of a *parent* or *super-component* together form the complete *parent* component [34]. Sub-components define non-overlapping regions of the component they form a part of for the stage in which they are defined; this allows only one component to be mapped to each region on the 2D and 3D models. Properties that hold true for any

²The web-based and downloadable EMAP browsers can be found at: <http://genex.hgu.mrc.ac.uk/Emage/database/emageIntro.html>

³The EMAP web site can be found at: <http://genex.hgu.mrc.ac.uk>

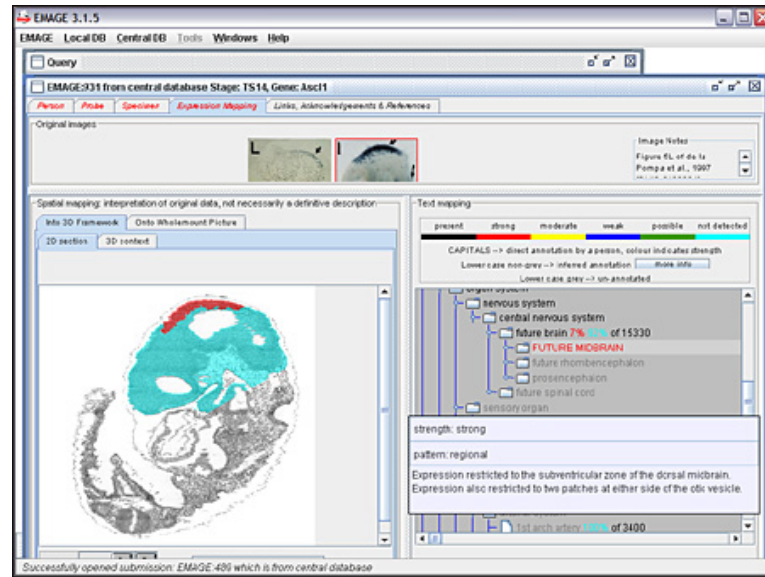


Figure 5.2. The results are shown for a search for the genes detected in three components in TS14. Red shows regions on the 2D slice where gene expression is detected, cyan where it is confirmed not to be detected, and grey for regions that have not been annotated. The text index also shows the percentage of each component for which gene expression is detected. Manual annotation is denoted using upper case lettering, and is displayed when the relevant component has the focus, as shown. Expression inferred from annotation uses black text for components along the path to the root from a component for which a specified gene is expressed.

component are propagated up the tree; these properties will also hold true for all ancestors of the component. Taking the *heart* as an example, if a gene *geneA* is expressed in the *left ventricle*, *geneA* will also be expressed in the *heart*. However *geneA* will not necessarily be expressed in the *right ventricle*, which is also a sub-part of the *heart*. Correspondingly, attributes not defined for any part of a component are also undefined for all its component parts. A second gene *geneB*, expressed in the *arterial system*, say, but not in the *heart*, will not be expressed in any of the component parts of the *heart*.

Figure 5.3 displays the ontology for TS04 using an indented text index and a graphical representation of its structure. Any non-leaf node may be collapsed, in which case it can be regarded as a composite node made up of its component parts. Node *B* represents the component *embryo* in TS08, with sub-parts *D* and *E* representing the components *compacted morula* and the *inner cell mass* respectively. *C* represents the *extraembryonic component*, *F* *cavities and their linings*, and so on.

Figure 5.3 shows only one of what could be several different visual representations of the relationships between components. Recognising that alternative anatomical structures could be derived based on other relationships identified between components, the concept of *grouping* (discussed in § 5.3.2 and § 5.3.4) has been developed [34]. *Grouping* allows explicit definition of implicit structures, by creating additional links between components that make up the structure(s) in question.

The *forelimb* and the *hindlimb* in TS26 of the mouse, for instance, both contain sub-components which are defined as *skin*. Alternative structuring of this data set could create

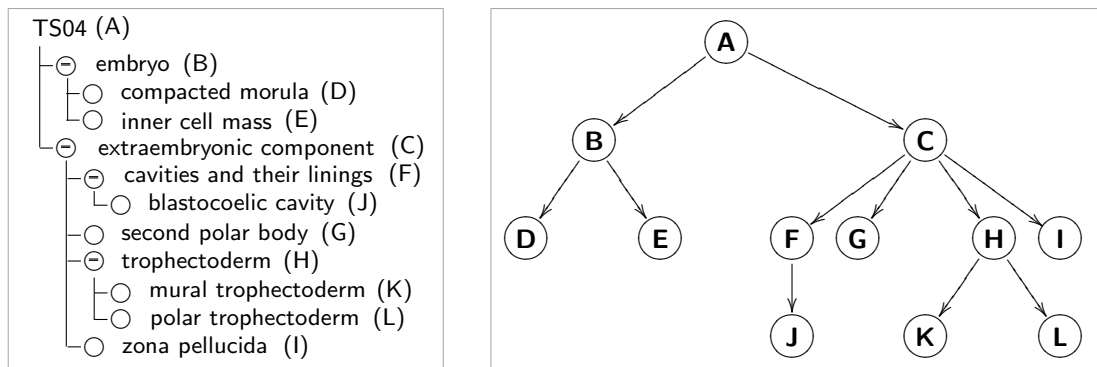


Figure 5.3. A node-link graph is used to reveal the hierarchical structure of the EMAP anatomy ontology for TS04 in the text index on the left, providing intuitive recognition of the *part-of* relationships between components.

a single (super) component *skin* which would link to all the parts of the *skin* in the mouse which currently form a part of other components. This would result in a modification of the default structure of the ontology, to reflect these new relationships.

5.1.2 Access and interoperability

The EMAP data is stored in an object-oriented (OO) database, ObjectStore^{TM4}. XML (the eXtensible Mark-up Language), providing a useful format for storing the hierarchically structured data, is one of the main options for presenting (the textual) ontology data. In addition to promoting data exchange XML has the advantage of being able to attach semantic content to the ontology data being studied.

Development of a large number of the EMAP tools using Java reduces problems with cross-platform compatibility and allows stand-alone use or online access from the EMAP web site. Providing web access to the data analysis tools developed for the EMAP project makes them easily and widely available, with the only restrictions being suitable hardware and network resources. [15] provide a description of the tools available on or for download from the EMAP web site).

5.2 The Cross-Species Anatomy Network

Evolutionary conservation in organisms results in similarity in coding regions of genes (for equivalent components). The XSPAN project⁵ maps gene expression data uncovered during the course of research to already identified structures in model organisms. These mappings may be extended to aid the identification of new genes, tissues and organs in related or even more distant species, to infer their structure and function. Information obtained may be used in genomics research, to study disease and other physiological conditions in humans, and in drug research and pharmacology. XSPAN is currently analysing relationships between five organisms using pre-existing anatomy ontologies:

⁴See the ObjectStore web site at: <http://www.objectstore.net>

⁵See the XSPAN web site at: <http://www.xspan.org>

- **Mouse**, using data from EMAP. Recognised as a model organism for the study of development in mammals, conditions identified in the mouse may be mapped to corresponding anatomical components and genes expressed in the human [85] and other mammals.
- **Human**, using the Atlas and Database of Human Developmental Anatomy⁶ at Edinburgh University's School of Biomedical Sciences.
- **Zebrafish**, using data from the Zebrafish Server at the Cardiovascular Research Center at the Massachusetts General Hospital⁷
- **Drosophila**, using FlyBase stored at the Indiana Genomics Initiative⁸.
- **C. elegans**, using data from the WormBase Consortium⁹.

Mappings that typically occur include equivalence based on function. The *heart* in each organism, for example, performs the same function, even if it has differences in component parts in different organisms. Common lineage, identifying the anatomical structures from which a specified component is descended, also serves as a point from which equivalence may be determined. This traces the paths that components follow across multiple stages of development, to differentiate between identical names that refer to different components and those that evolve from equivalent structures in an organism (see § 5.3.3). Common cell type may also be used to determine equivalence between components. The XSPAN web site¹⁰ explains the different methods used to create mappings between components in distinct organisms. Figure 5.4 shows three types of equivalence identified between components in the *mouse* and *Drosophila*.

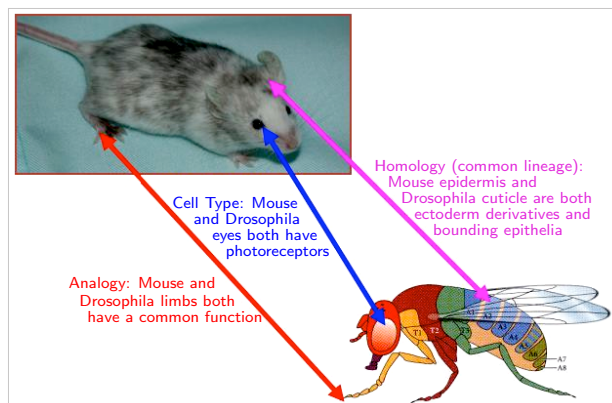


Figure 5.4. Identification of similarity in anatomical components in the *mouse* and *Drosophila* (Image used with permission from: http://www.xspan.org/technical/expert_mapping.html)

Difficulty analysing data from multiple sources has been previously discussed, the most important including data storage using incompatible data types and database schemas [19].

⁶See Humat at: <http://www.ana.ed.ac.uk/anatomy/database/humat>

⁷See the Zebrafish Server at: <http://zebrafish.mgh.harvard.edu>

⁸See FlyBase at: <http://flybase.bio.indiana.edu>

⁹See the WormBase Consortium at: <http://www.wormbase.org>

¹⁰Information on expert mappings in XSPAN can be found at: http://www.xspan.org/technical/expert_mapping.html

Specific difficulty in comparison between ontologies includes different methods for describing equivalence or other relationships between elements [31], and varying definitions for the same or similar terms [155]. However, research has shown the potential for improved analysis of data stored using (standardised) ontologies. XSPAN makes use of anatomy ontologies to provide a reference framework that aids the determination of equivalence across components in multiple organisms, based on similarity in gene expression and cell type, or expert opinion. Mappings may also use lexical analysis to match components, based on component names. This should help to resolve poor interoperability between different data sources [15], common to biological and other scientific data. The research on data integration in the XSPAN project is also related to work being done by the SOFG, with the creation of the SAEL [136] (refer also § 4.1.1).

5.2.1 Access and interoperability

The different sources of data for the anatomy ontologies of each of the model organisms are fed into a common data warehouse built using IBM's DB2^{TM11}, from which queries may be formulated to retrieve similarity between components across different organisms. Equivalence determined between components across different ontologies may then be mapped to gene expression in corresponding data sources such as the GXD. Figure 5.5 shows how the XSPAN data warehouse is used to integrate the multiple data stores used in the research project and communicate with the end user, to improve translation of terms between the model anatomy ontologies, and aid identification of equivalent components in the different organisms being studied.

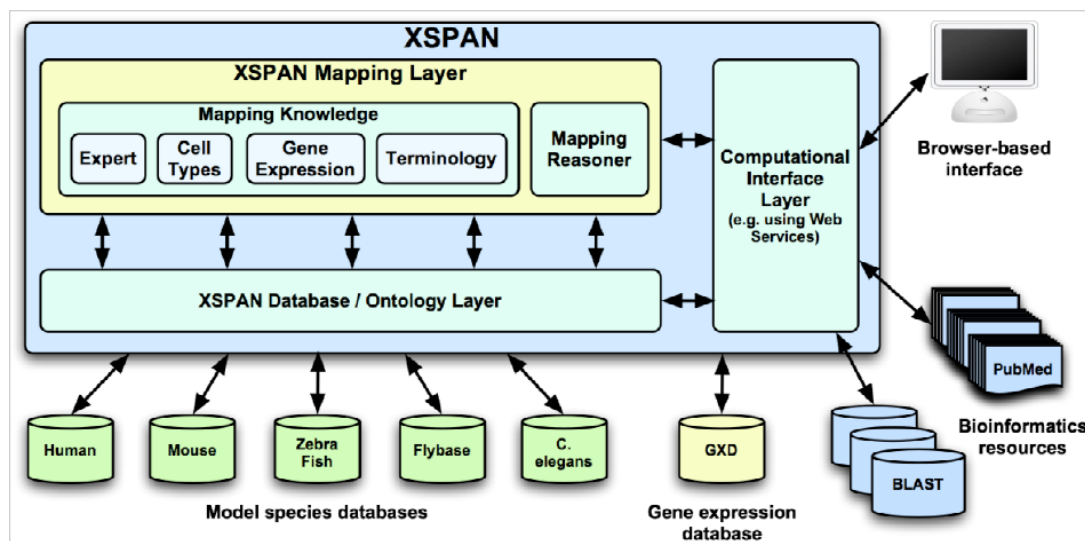


Figure 5.5. Data from model anatomy ontologies and the GXD is fed into the XSPAN data warehouse, the interface to the query system and the web-based tools used to record information on equivalence between components, based on domain knowledge and current research. (Image used with permission from: <http://www.xspan.org/overview/index.html>)

¹¹The IBM DB2 web pages can be found at: <http://www-306.ibm.com/software/data/db2>

To increase accessibility and promote interoperability web-based analysis tools provide an interface to the ontology data. XML improves data exchange, allowing automated data analysis. Natural language processing is also being used to aid analysis of terms found in the different ontologies. Figure 5.6 shows the structure of the XSPAN prototype, illustrating data flow through the system.

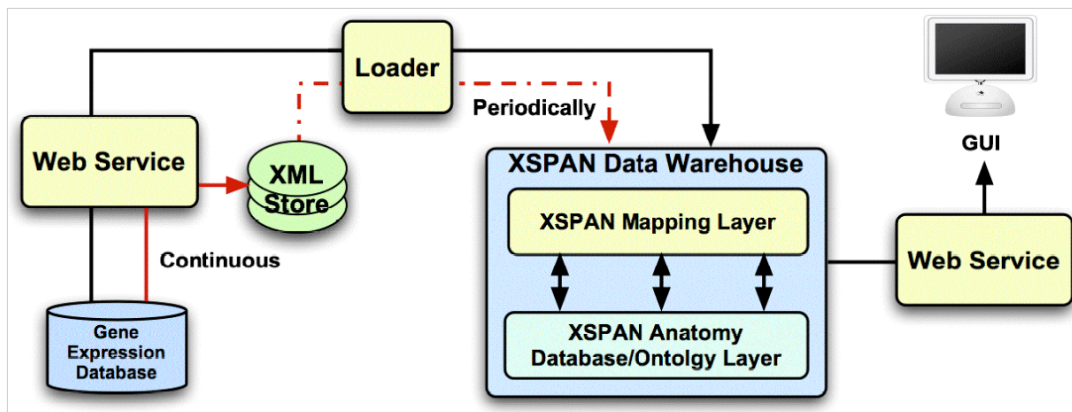


Figure 5.6. Structure of the XSPAN prototype, illustrating data flow
(Image used with permission from: <http://www.xspan.org/overview/details.html>)

5.3 Data analysis requirements for EMAP and XSPAN

5.3.1 Recognition of data structure

Although the indented text index provided in the EMAP section browser gives some indication of the hierarchical structure of the ontology data it is difficult to obtain an overview of any but the very small data sets, as figure 5.7 illustrates.

Relationships between especially widely separated elements are also difficult to recognise. This can be seen for even the relatively small dataset for TS04 — with only three levels of nesting relationships among data elements are not easily recognised using the text index. Mapping the structure of the data to a hierarchical visualisation as in figure 5.3 would provide an overview that highlights relationships in the data. The advantages a graphical representation would provide for TS26 are obvious; much deeper nesting of data combined with the large number of elements in TS26 makes it even more difficult to determine the structure of the data set.

5.3.2 Alternative structuring of data

The predominant relationship found between components in the anatomy ontologies under study is *part-of*, such that each set of sub-components forms a complete *super* component. It is however recognised that alternative classification of data may result in relationships other than those defined in the default structures presented. As an example, [33] describe how a *group* node could be created to represent the complete structure *skeleton* in the

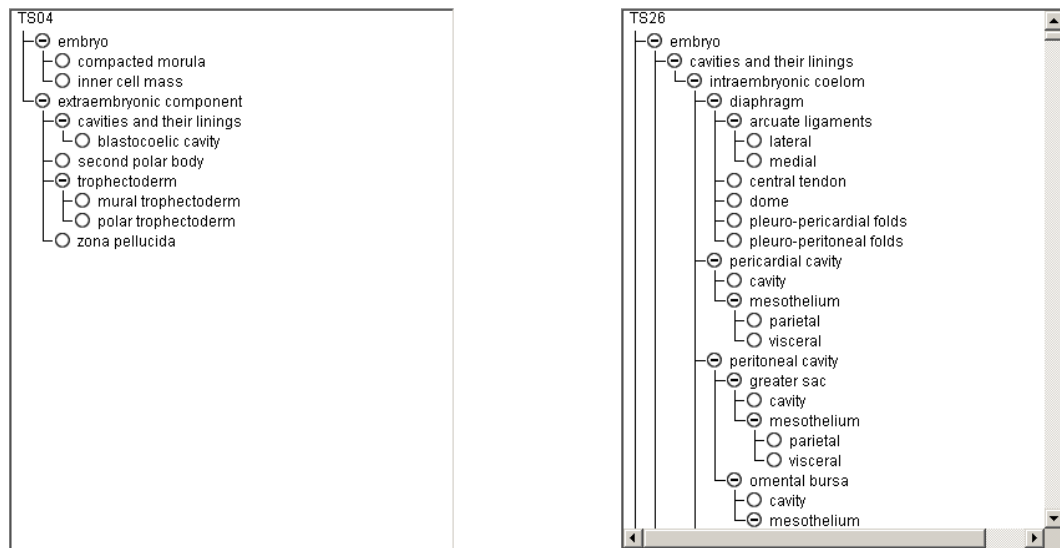


Figure 5.7. The indented text index that forms part of the EMAP section browser is used to display the anatomy ontologies for TS04 and TS26. The structure of TS04 is easily discerned as all 12 components fit comfortably in the text area on the left. It is more difficult to obtain a good mental model of the structure of TS26; with all nodes expanded a large amount of scrolling is required to view all 1749 components it contains, as the relatively small scroll button shows.

mouse embryo, composed of the individual sub-components that each represent a part of the skeleton, but which, in these ontologies, form *part-of* relationships with other named structures. Further, deriving relationships based on criteria other than structure, using function, for instance, provides additional options for structuring the ontologies [34].

Graphical support is required for *grouping* components based on user-specified criteria, to reveal alternative structures such as described in figures 5.8 and 5.10. § 5.3.4 provides more detail on the impact of *grouping* on the default structure of each ontology.

5.3.3 Tracing lineage within data

Lineage across stages of development traces a component from the point where it first appears till it develops into another component or ceases to exist. The EMAP browsers currently trace lineage using a sequential arrangement of up to 28 text boxes along a horizontal plane, each representing a stage of development, and containing the list of components that occur within that stage (see figure 5.9). Beyond a very small number of stages scrolling is required to navigate through the data; a visualisation that presents an overview of all data would provide a more intuitive method for mapping the paths desired, and reduce the cognitive (memory) load associated with the current method for browsing the data.

5.3.4 Representation of complex relationships

Individual stages of development in the mouse comprise a list of unique components, each of which forms a *part-of* relationship with its parent. All sub-components of a node together form a complete (super) component. The default structure of the ontologies being studied

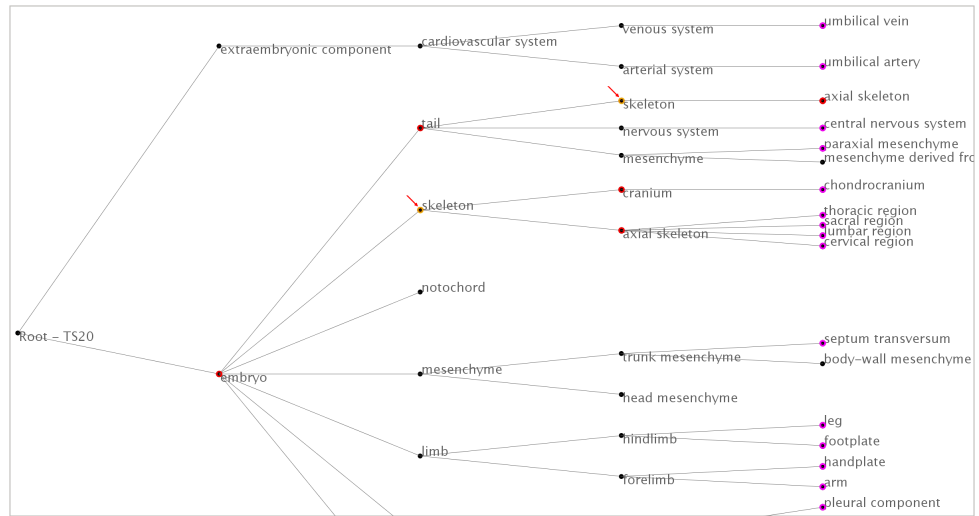


Figure 5.8. The 2D browser developed as part of this project is used to visualise TS20. The component *skeleton* appears three times: the two instances that form sub-parts of *embryo* and *embryo.tail* are highlighted. In order to *group* all components that make up the *skeleton* a *group* node could be created with *part-of* links to the sub-parts of each distinct *skeleton* node and to their parent components.

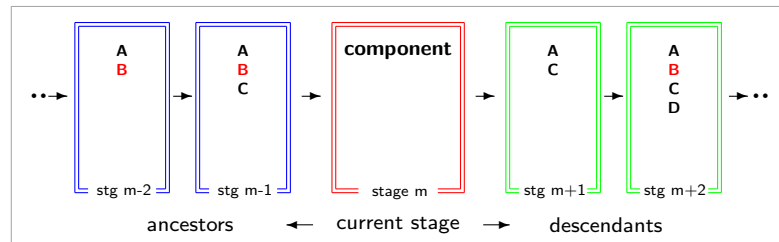


Figure 5.9. The box diagram illustrates the method currently used to trace lineage in EMAP. A large amount of scrolling is required for components that span more than a few stages of development, resulting in a large cognitive memory load.

restricts inheritance in the tree to a single parent, so that only one path can be drawn from a component to the root. Alternative structuring of data to form *groups* may however result in multiple parentage for data components and, thus, multiple paths to the root [15], as illustrated in figure 5.10.

Different classification methods may mean that a component *K* may be seen to form *part-of* a structure *A*, based on *classification 1*, and a part of structure *Y*, based on *classification 2*. The tree becomes a DAG, and propagation of component attributes up the tree (as discussed in § 5.1.1) no longer holds for all paths to the root. However non-existence of an attribute anywhere in a component still propagates downwards, provided all descendants follow only one path to reach the component queried.

The *coronary artery*, for instance, forms a part of the *arterial system*. However the *coronary artery* could be seen to form a part of the *heart*. If the gene *geneB* expressed in the *arterial system* is also found to be expressed in the *coronary artery*, then even though *geneB* is not expressed in the *heart*, *geneB* would still be expressed in the *coronary artery* (which is now seen to form a part of the *heart*).

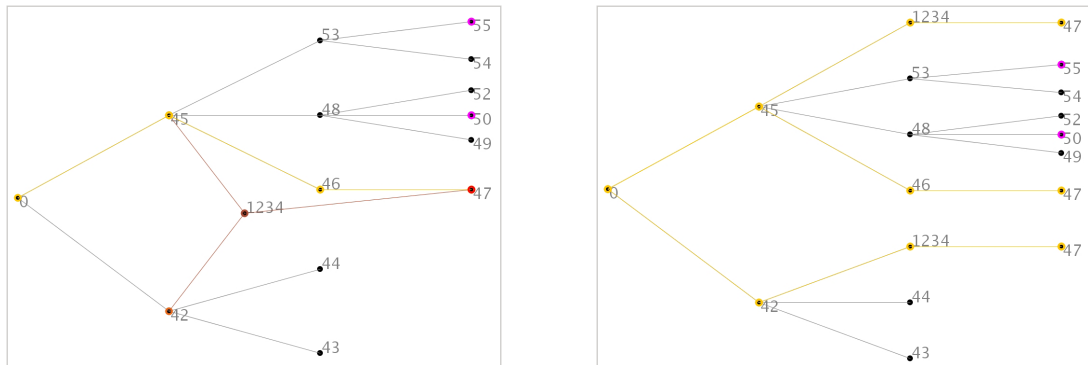


Figure 5.10. A *group* node *1234* is added to the tree representing the ontology, with links to two parents *42* and *45*, and a child, *47*. The default path to the root from *47* is traced in yellow. Cloning the *group* node on the right maintains a pure tree and the three paths *47* now traces to the root are highlighted. For a large tree cloning components would contribute to the problem of occlusion common to hierarchical graphs. Additional visual cues would also be required to ensure cloned nodes are not overlooked, so that data structure is properly interpreted.

Persistence of components across stages may result in repeated entries when all stages of development are merged to form the *abstract organism*. Unique IDs however differentiate components with the same (fully qualified or simple) name in individual stages of development and in the *abstract organism*, allowing independent use of the different data sets. Figure 5.11(a) highlights the components *second polar body* and *zona pellucida*, which first appear in TS01, where they form two of four sub-components of the root. The two components persist through to TS04 (see figure 5.11(c)), where they form sub-components of the *extraembryonic component*. Figure 5.11(b) shows where the *compacted morula* first appears in TS03, also as a sub-component of the root. Like the two components previously identified it persists through to TS04 where it becomes a sub-component of *embryo*. The graphical representation of the *abstract mouse* shown in figure 5.12 therefore has two copies of each of these nodes, one set as sub-components of the root, and the other as sub-components of the *extraembryonic component* and the *embryo* as applicable.

5.3.5 Visual, dynamic querying

The EMAP section browsers provide simple text searching, mapping the first hit found in the component index to corresponding regions on 2D slices of the 3D models of embryos (refer § 5.1 and figure 5.1). Repeating a query cycles through the component list, successively returning the next match found. The method of searching is, however, fairly tedious, especially for a frequently occurring substring.

An alternative to searching in the EMAP browsers is to browse through a drop-down list displaying fully qualified names for each component. However except for very short lists this may place a large cognitive load on users.

Ability to retrieve all data satisfying search criteria in a single request would provide simpler and more intuitive searching, especially where comparison within search results

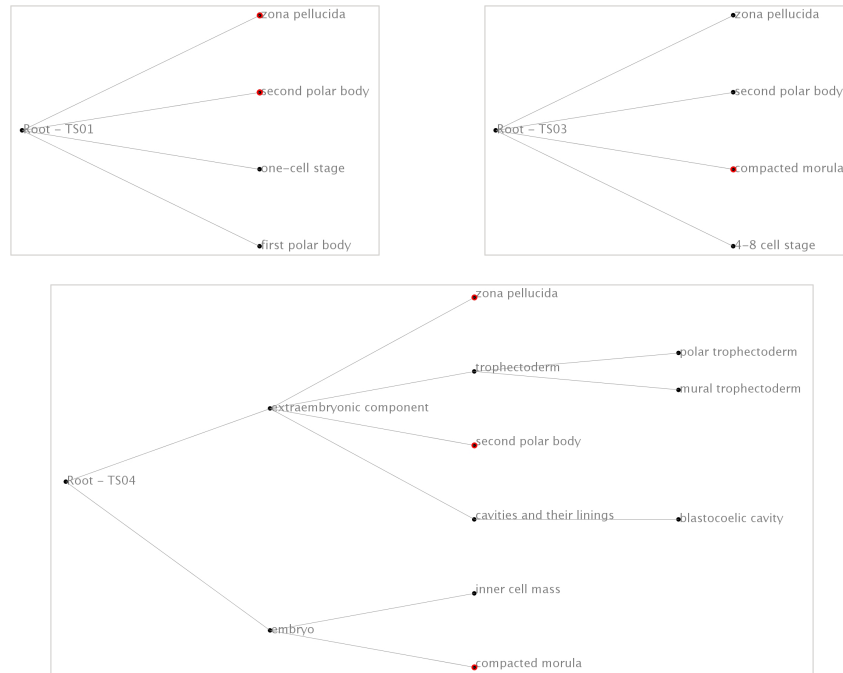


Figure 5.11. The snapshots at the top highlight three components where they first appear, in TS01 and TS03, during the development of the mouse embryo. The bottom figure shows the last stage in which they occur; the components identified are no longer sub-components of the root, but lie on the next level in the tree.

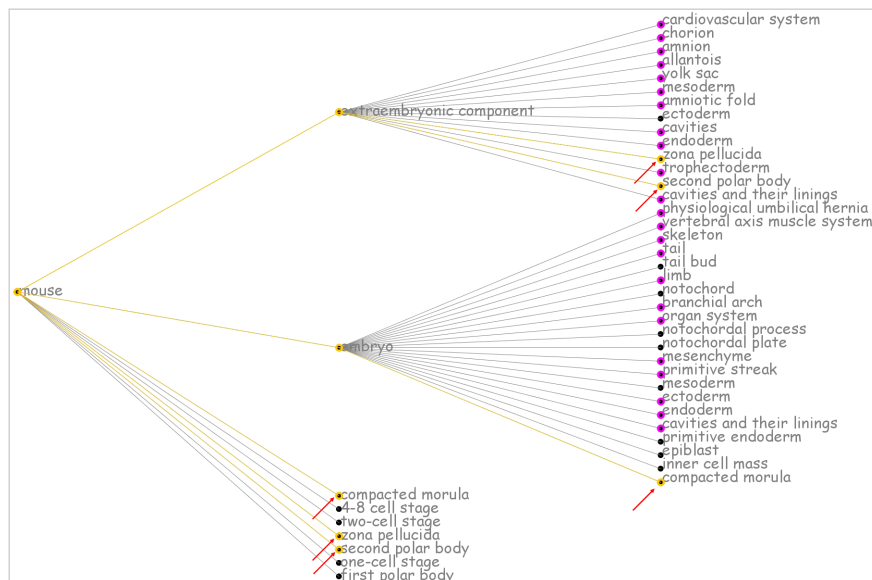


Figure 5.12. The first three levels of the graph for the *abstract mouse*, which contains all instances of components occurring during the individual stages of development of the embryo. Each component highlighted occurs twice, at different levels in the tree, corresponding to positions relative to the root in the distinct stages of development identified in figure 5.11.

is required. Options for visual, dynamic querying as discussed in § 2.4 would provide additional perceptual cues to aid the formulation of effective queries. Visual querying is also able to provide users with cues both on search hits and on data not meeting search criteria, useful where extension or modification of queries is required. Allowing users to search from a component, using data attributes to provide search criteria, helps identify effective keywords and search terms.

The EMAP browsers currently provide the option to extend searching for corresponding gene expression data to the EMAGE database and the GXD. Further options that extend querying to other (previously verified) data sources would remove the burden of locating reliable sources from the user. Functionality that manages integration between different data sources would also allow simultaneous querying of multiple data sets, further enriching information retrieved. Aids for natural language querying and transparent reformatting of queries to suit underlying database schemas would be beneficial to both expert and non-technical users. Additionally customisability that allows complex queries to be performed using formal syntax would provide advanced IR and give expert users more confidence in information retrieved.

5.3.6 Multiple, simultaneous analysis of ontologies

In order to trace lineage across multiple stages of development in a single organism, or compare components across organisms for equivalence, it is necessary to perform simultaneous analysis of multiple ontologies. [32] illustrates how lineage in one species can be inferred from previously determined lineage in another species, based on spatial or other mappings between equivalent components in the ontologies to which each component of interest belongs. Figure 5.13 shows how graphical support for analysis of multiple data sets could be used to highlight relationships that occur within and between data. Similar structures could be used to infer lineage, by using (physical) mappings to represent the lineage relationships.

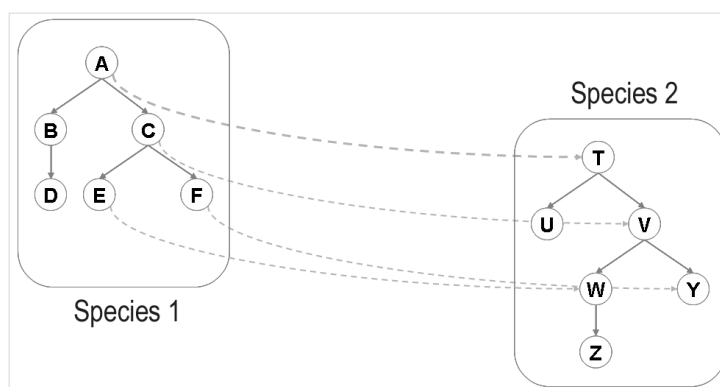


Figure 5.13. Component *C* forms a *part-of* the root component *A* in *Species 1*, and is itself composed of two *parts* *E* and *F*. The root of *Species 1*, *A*, maps to the root *T* of *Species 2*, and component *C* maps to *V*. Based on similarity in structure between the two data sets it may be inferred that the component parts of *C*, *E* and *F*, map to *W* and *Y* respectively, in *Species 2*.

5.4 Graphical analysis of bio-ontologies

Ontologies will normally be hierarchically structured, so that they lend themselves well to graph visualisation. This simple visualisation method provides intuitive navigation and exploration that highlights relationships within and across data sets. Visualising the anatomy ontologies under study using hierarchical graphs should help to reveal the relationships that occur between different components within individual and between different, but related organisms.

5.4.1 Assessment of existing graphical analysis techniques

§ 2.5.1 discusses the importance of modular development of components and tools, to allow extension and customisation that satisfies individual user preferences and analysis requirements. [47], among others, have found however that very few tools are built so that they can be easily integrated with other systems. It is even more difficult to incorporate multiple, independently developed tools into a single system, to allow the varying functionality and techniques each provides to be used in concert for analysis and IR. Further, tools are often built to suit a specific set of requirements, so that they are not very effective for analysis that does not map directly to the problems they are intended to solve.

This section reviews the tools examined in chapters 3 and 4, to determine which could be used without modification for the analysis required, and to identify existing functionality which may be extended to satisfy better the data analysis requirements identified. (Although the following sections group tools by the type of visual analysis provided, it should be noted that some of these tools fall into more than one of the categories named. What is seen to be the strongest feature of each tool is used as the main classification criterion in such cases.)

Hierarchical graph visualisation

A number of phylogenetic tree drawing tools were examined, including the *PHYLIP* suite, *Phylo dendron* and *ATV Viewer*. The main attraction of these tools is the hierarchical graph visualisation they provide, which could be used for simple but effective visual analysis of the ontology data being studied. However a major limitation in the applications examined is poor scalability, being able to draw only a few hundred nodes before occlusion begins to degrade usability. The data sets of interest in *EMAP* and *XSPAN* range from less than 10 nodes to almost 2000 in a single stage of development, to over 3500 in a single, abstract organism, so that this poses a significant problem. A contributory factor to low scalability is the large amount of wasted screen space inherent in tree graph visualisation. This is illustrated for *ATV Viewer*: although over 500 nodes can be drawn at once, occlusion makes it difficult to visualise more than a relatively small number of nodes, while portions of the display remain empty (see figure 3.7). *ATV Viewer*, however, provides functionality for redrawing sub-trees at maximum magnification, resolving occlusion for ROIs, but with

the loss of surrounding context; no visual cues are available to indicate position in and relevance of an ROI to the rest of the tree.

Although able to visualise a very large number of nodes (in multiple trees) effectively, *TreeJuxtaposer* still suffers from occlusion typical of tree graphs, with the main source of clutter being node labels; high density of data in the 2D graphs used in *TreeJuxtaposer* may reduce users' ability to identify data of interest. To counter this, labels that would occlude previously drawn data may be suppressed, while labels for data of interest are highlighted by varying the colour used to draw them.

Another hierarchical visualisation tool assessed was *SpaceTree*. User requirements in [139] for focus on detail in ROIs, however, restrict visualisation for trees with deep nesting such as the ontology data being studied. Only a few levels in a tree are displayed on the screen at a time, making it difficult to obtain a complete overview of the structure of very large data sets or deep trees (refer figure 3.3, where only a relatively small subset of the 199 nodes in the data set are displayed). Another limitation is that sub-trees other than those of immediate interest are collapsed; although this allows detailed analysis of ROIs it is not possible to compare especially widely separated elements in the tree.

uDraw(Graph), on the other hand, provides complete overviews of data sets in addition to abstraction that collapses sub-trees into composite nodes, or fades away less important data, to deal with complexity in the overview. *uDraw(Graph)* is able to load multiple data sets simultaneously, a requirement for tracing lineage and for determination of equivalence across ontologies. However each data set is loaded in a different window, so that it is necessary to map between multiple screens, placing a large cognitive load on users and preventing truly simultaneous analysis of multiple data sets.

Tools provided for management of GO and other similar ontologies generally provide very simple graph visualisation, with limited functionality for analysis and interactive modification of layout.

The visualisation plug-ins provided for use with *Protégé* present fairly sophisticated functionality for visual interaction with ontologies. *Piccolo*'s continuous zoom that provides F+C in *Jambalaya* is useful for reducing the disorientation that occurs during navigation and exploration of data. Functionality for encoding data attributes, different options for data layout and the ability to save analysis sessions for future or continued use all help to reduce complexity in data analysis. It should be possible to extend *Jambalaya* to provide intuitive graphical support for creation of *groups* in 2D (as described in § 5.3.2).

OntoViz makes use of non-standard programming languages for drawing graphs. The scope of this project and a preference for any new tools developed to be easily integrated with existing tools in EMAP and XSPAN mean that *OntoViz* may not be a good choice for developing further functionality for visual analysis.

Scatter plots and network graphs

Maintaining a stable mental model of data structure is important if relationships inherent in data are to be recognised intuitively. A disadvantage in *TouchGraph* is one inherent to spring layouts: multiple runs of the algorithm may produce different layouts. Coupled with the constantly changing structure of the visualisation during navigation it is difficult to develop and maintain a stable mental model of data structure.

BioLayout provides a stable layout, and graphical support for classification of data that would be especially useful in creating *groups*. Occlusion due to a large amount of interlinking between nodes and overlap of nodes in areas of high density may, however, prevent different relationships from being properly distinguished.

J-Express makes use of scatter plots based on PCA for data visualisation. The research done to this point has however found that hierarchical graph visualisation probably presents an optimal solution for visual analysis of the anatomy ontologies under study. Further, having developed into a commercial tool, modification and use of *J-Express* are naturally restricted.

2D hyperbolic layouts

The use of 2D hyperbolic layouts in *OntoRama* and *HyperGraph* allow the display of much larger data sets and with far less occlusion in ROIs than would occur in equivalent Cartesian layouts. *OntoRama* also maps data in a hierarchical text index to the graphical display, and could thus provide a familiar interface for current users of the EMAP browsers. However the constantly changing layout of data during navigation and the distortion that is a residue of hyperbolic layouts prevent the formation of a stable mental model of data structure in the two applications.

3D and VRML

A limitation in the 2D tools analysed is the occlusion that occurs beyond a fairly low threshold. Effective abstraction of data can be used to provide useful analysis of individual data sets. It would however not be possible to visualise multiple ontologies simultaneously in any of the 2D visualisation applications reviewed such that the analysis and comparison of individual data elements and their attributes could be obtained as required. A solution to this problem may be to use 3D, with the larger amount of space it contains, for multiple, simultaneous visualisation of the anatomy ontologies being studied.

Tools examined that make use of VR and VRML, such as *VRMLgraph*, are able to take advantage of built-in aids for navigation and exploration in VR browsers, and the larger number of degrees of freedom for navigation available in 3D. *VRMLgraph* would be useful for generating tree structures in 3D, but requires additional functionality to obtain the interactive analysis required for this project.

Use of VRML also requires the installation of dedicated viewers or plug-ins within web browsers, an extra burden for users, especially where low-end systems are in use. Further, to take full advantage of the benefits of virtual worlds additional hardware and software that provide haptic feedback and stereoscopic vision and sound are often required. This is however not normally possible for the desktop computing available in typical users' normal working environments.

Cone trees provide an alternative solution with the compact visualisation they provide. However the structure of the trees generated would result in crossing of links if used to visualise relationships crossing data sets, making it difficult to recognise relationships identified.

The main advantage in *Walrus* is the ability to visualise hundreds of thousands of nodes in the 3D hyperbolic layout before occlusion becomes a problem. However *Walrus* was developed as a standalone application, and it cannot be incorporated into other tools to enable use in conjunction with other modules. Also, an additional layer is required to convert input to the non-standard *LibSea* file format it uses. Once loaded, graphs generated in *Walrus* cannot be modified; one requirement for analysis is however the dynamic creation of links between data elements to represent new relationships discovered, in addition to other requirements for interactive extension to or modification of graphs. The ability to load only one graph at a time also means that it is not possible to compare multiple data sets in *Walrus*, a requirement for mapping relationships crossing the anatomy ontologies being studied.

5.5 Proposal for a solution for data analysis

The range of tools evaluated provide different functionality that satisfy some of the requirements for analysis detailed in § 5.3. The simple abstraction offered by node-link graphs should provide a useful method for visualising the anatomy ontologies being studied. However limitations in space in 2D mean that functionality is required that can generate data overviews while minimising occlusion in the graph.

SpaceTree provides very useful functionality for analysis of ROIs, but would require modification of the interpretation of the ontology data being studied in order to make optimal use of the cues provided for encoding hidden data. *Jambalaya* provides visualisations with functionality that most closely approach requirements for the presentation of overviews of individual anatomy ontologies. Encoding of data attributes using size, shape and colour of data nodes should be useful for describing data attributes. Additional graphical support is however still required for creation of *groups* of nodes based on user-specified criteria, to reveal alternative structuring of data sub-sets without destroying the default hierarchical structure of the data.

The compact visualisations provided by *treemaps* provide a potential solution to the space limitations encountered in the use of node-link graphs. [11] demonstrate use of the technique for visual analysis of GO data, with functionality for dynamic querying that is able

to link to and retrieve information from external data sources, providing a potential solution to the analysis required for individual data sets. However the simultaneous visualisation and comparison of multiple ontologies required for this project cannot be satisfied using this technique, as the layout does not make it possible to compare relationships among distinct data elements directly, especially across multiple data sets.

Support for visualisation and analysis of multiple ontologies is required that highlights relationships both within individual data sets and equivalence across ontologies. Ability to map paths of interest within single and lineage across multiple hierarchies, tracing persistence of a component through different stages of development is also necessary. Hyperbolic layouts were considered as an option for the display of the large amounts of data involved. However the distortion of such layouts and constant change in data structure associated with the F+C technique that is an important element in analysis of ROIs mean that hyperbolic visualisation may not provide an optimal solution.

The larger amount of space available in 3D provides an alternative solution for the occlusion that occurs due to limited space in 2D. However 3D visualisation comes with problems of its own. Objects closer to the viewpoint still occlude more distant elements; it is necessary to rotate visualisations or move around objects to view others hidden behind them. Disorientation during navigation commonly occurs, especially when users become immersed in local areas of the virtual world created and the context of the overview is lost. Typical solutions to these problems include the provision of landmarks and multiple cameras or viewpoints that serve as reference positions. History sessions also allow users to return to previous locations or views. Reduction to 2.n dimensions, (n between 0 and 5) lowers immersion in data and allows users to *fly over* what approaches the landscape that [39] describes, to regain the context of the overview.

No single tool was identified that would be able to provide the support required for visual analysis of multiple ontologies, to trace lineage within an organism or map equivalence in different organisms. Another important consideration is the need to provide extensions that can be easily integrated into the EMAP browsers in current use. Ability to provide online use would also be an advantage, as it removes the burden of installation of new tools and data management from the user. Update of tools is also simplified as a single, central point can be provided for distribution of extensions and updates.

Restrictions in time and scope of this project mean that it would not be practical to first learn how to customise and extend an existing tool and then develop a second tool to provide what would be novel functionality to satisfy the additional requirements of this project. It was therefore decided to develop an independent tool that builds on existing functionality for visualisation of individual ontologies in 2D, incorporating different techniques identified in the tools examined that provide useful options for analysis. A decision was made to limit visual analysis to 2D for single data sets, as techniques available for abstraction and analysis of ROIs should work reasonably well for the data set sizes involved (up to 2000 elements

for a development stage in the mouse, for instance, and just over 3500 for the *abstract mouse*). Anecdotal evidence, supported by research, also suggests that 2D visualisation may be better suited (than 3D) to the 2D display surfaces that are used in the typical target user's normal working environment, in addition to providing a more familiar and less complex interface for presenting information to users [39, 78].

Development of a single tool incorporating multiple analysis techniques would also provide the opportunity to evaluate existing techniques for visual analysis. Following this a 3D visualisation system would be developed incorporating novel functionality to resolve outstanding issues in visual analysis of the anatomy ontologies, to provide intuitive identification of relationships across multiple data sets.

5.6 Summary

Ontologies serve as knowledge bases; use may involve simply searching within a single ontology to identify relationships between elements it describes, or the retrieval of specific information. Different ontologies may describe related information, in which case analysis may look at determining mappings between individual elements in each ontology, to retrieve similarity within what are often independently created knowledge sources, to enrich existing information and/or retrieve new information.

Myriad data analysis solutions exist, most of which focus on specific fields and/or types of data (refer chapters 2 and 3). This thesis looks at the comparison of ontologies, to retrieve similarity in related data sets. Focusing on a specific research area or data sub-set limits the scope of research and analysis so that intuitive analysis solutions may be developed for user information requirements in the field.

This chapter identified typical requirements for analysis and information retrieval in research that involves the use of anatomy ontologies. These start with the need for overviews of data that aid users in constructing effective mental models of data structure, followed by the ability to determine relationships within individual data sets and that span multiple ontologies. An important requirement is to provide intuitive methods for analysis that minimise the cognitive load on users; research in information visualisation and anecdotal evidence point to the advantages in harnessing highly advanced perceptual ability in humans in order to improve the process of data analysis and lead to results that more closely match user requirements. A review of existing analysis tools and techniques (with a focus on visual analysis) in this and previous chapters determined where current solutions may provide the analysis required, and where they are unable to satisfy user requirements fully.

In order to be able to evaluate different analysis solutions with typical target users it was necessary to build a prototype implementing existing and developing novel techniques for data analysis and IR. Chapter 6 details design of the two prototypes built to assess the alternative data analysis solutions proposed, and to examine how these solutions could be

extended to provide analysis of other similar data sets.

Chapter 6

Developing solutions employing visual analysis

The research done to this stage looked at data analysis and different techniques developed to provide intuitive options for analysis. This thesis focuses on harnessing advanced perception in humans to reduce cognitive load in especially complex data analysis; chapters 2 and 3 discuss the advantages in visualisation of information, namely, increased ability to obtain an overview of data structure and identify patterns and relationships in data. Existing solutions were not identified that would be able to satisfy fully, the specific information requirements of the user group studied (refer § 5.3). The main limitation of current bioinformatics and other data analysis tools is the focus on a relatively small sub-set of data, normally to prevent cognitive overload in analysis or because of the large amount of computing and other resources required to support processing and analysis of large data sets. The analysis required for this project, however, involves comparison of multiple data sets, employing detailed analysis of ROIs and overviews of data structure that highlight relationships spanning ontologies.

This chapter describes the development of an application incorporating existing analysis techniques, starting first with simpler 2D analysis, to allow evaluation of these options when used in concert. This is followed by the design of alternative, novel options for visual analysis, to provide solutions for user information requirements that cannot be fully satisfied by existing methods. This involves an extension to the third dimension to make use of the larger amount of (virtual) space available for holding data. An analysis of user information requirements and prior experience with other visualisation systems fed into the system design, using ontology data in EMAP to test the options developed for analysis. Other design considerations included the ability to integrate new tools with systems currently available for analysis of similarly structured data, in order to increase the learnability of new analysis techniques developed. Designing for extensibility, to allow analysis and comparison of other similarly structured ontologies, was also an important factor in developing and assessing the new techniques developed for visual analysis.

To ensure the new solutions proposed would map to the requirements of target users and that they would provide usable and improved analysis options, an empirical, user-centred cycle was used to guide development of the prototypes built to evaluate the different options for analysis. A difficulty was obtaining a sufficient number of target users so that the conclusions drawn from the evaluations performed could be validated based on statistically significant results. The alternative, reliance on expert review of analysis techniques developed, however, has the potential of being biased by personal opinion or misunderstanding or incomplete understanding of users' information requirements or work environments. Further, results of research may not map to use of technology in the workplace; testing new ideas with actual users is important to minimise such differences. Figure 6.1 shows the cycle used to guide research, development and evaluation, seeking improved options for analysis.

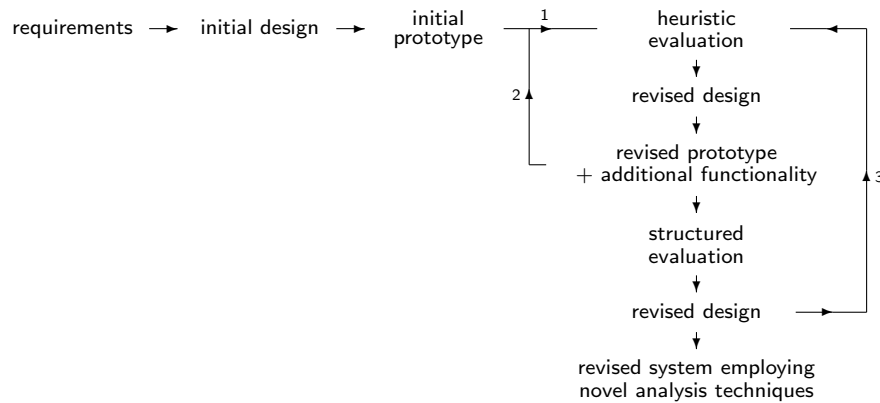


Figure 6.1. Empirical, user-centred cycle followed in development and assessment of novel approaches to visual analysis

6.1 Choice of programming language

Development with an aim to augment research with minimal disruption to users had a significant influence on the choice of development environments and languages considered for building a new visual analysis system. EMAP tools currently available for analysis of the ontology data are built mostly using Java, largely to provide cross-platform access to resources, and web access and use where practicable. These considerations played a large role in choosing to develop the analysis solutions for this project using Java.

6.2 Data types and storage methods

An important requirement of the visual analysis application to be developed was the ability to input data in different formats and from different sources and also output data in forms that promote exchange and reuse. The following sections describe the different methods used to store, and options available for exporting the EMAP data.

6.2.1 Database

The Common Object Request Broker Architecture, CORBA, acts as a server, providing a layer between the EMAP and EMAGE databases and a large number of the tools used for data analysis. Java Servlets are also used to provide a cross-platform interface between the gene expression data and EMAP tools. Using Servlets also has the advantage of allowing easy dissemination on the Web, increasing access to data and analysis tools.

6.2.2 XML

Previous sections (refer § 2.2.1 and § 2.4) have discussed the importance of data access and exchange using standardised formats and a common language. XML, with its self-describing syntax, is a good choice for storage of regularly structured data [185]. Being both human and machine-readable using XML to store and export data eases both manual and automated analysis. Figure 6.2 shows an extract from the XML file used to describe TS11 for the mouse embryo.

6.2.3 Image

Each anatomical component stored in the text indices is mapped to a corresponding area on a 2D slice and 3D region cut out of reconstructed models of the mouse embryos, as described in § 5.1. This provides a spatial representation that aids analysis of the textual data. The EMAP site also provides a pictorial index¹ showing a snapshot of the mouse embryo at a specific point during each stage of development, each linked to a description of development during that stage and a plain text index (also using indentation to reveal data structure) that lists component names and EMAP IDs.

6.3 Structure of application

6.3.1 Designing for modularity and extensibility

To promote (re)usability and extensibility a major feature of the application design was separation of functionality for generating the visualisations from data fed into the application. This was to allow visualisations to be generated independent of data source and format. A layer between these two parts of the application would be used to identify data source, type and/or format, parse the input as required, and generate visualisations based on data content.

6.3.2 The data access layer

For each data type and/or format used to store the anatomy ontology data it is necessary to write a **Loader** that will read the input based on file structure and content, build objects

¹See: <http://genex.hgu.mrc.ac.uk/Databases/Anatomy/Diagrams> (last viewed Jul 2006)

```

<?xml version="1.0" standalone="yes" ?>
<HGU_MRC_Edinburgh>
<date>7/8/2001</date>
<species>mouse</species>
<anatomy><stage name="TS11"></stage>
  <component name="embryo" id="147">
    <printName>embryo</printName>
    <abbreviation></abbreviation>
    <childrenId>169</childrenId>
  ...
  <childrenId>163</childrenId>
  <startEmbryoStage>04</startEmbryoStage>
  <stopEmbryoStage>26</stopEmbryoStage>
  <parentId>0</parentId>
    <component name="cavities and their linings" id="148">
  ...
    </component>
    <component name="yolk sac" id="202">
      <printName>extraembryonic component.yolk sac</printName>
      <abbreviation></abbreviation>
      <childrenId>204</childrenId>
      <childrenId>203</childrenId>
      <startEmbryoStage>10</startEmbryoStage>
      <stopEmbryoStage>12</stopEmbryoStage>
      <parentId>176</parentId>
        <component name="endoderm" id="203">
  ...
        <component name="mesoderm" id="204">
          <printName>extraembryonic component.yolk sac.mesoderm</printName>
          <abbreviation></abbreviation>
          <childrenId>205</childrenId>
          <childrenId>206</childrenId>
          <startEmbryoStage>10</startEmbryoStage>
          <stopEmbryoStage>11</stopEmbryoStage>
          <parentId>202</parentId>
            <component name="blood island" id="205">
  ...
            <parentId>204</parentId>
            </component>
          </component>
        </component>
      </component>
    </anatomy>
  </HGU_MRC_Edinburgh>

```

Figure 6.2. Extract from the EMAP XML file describing TS11 of development of the mouse embryo. The full listing for TS11.xml can be found on the EMAP web site at <http://genex.hgu.mrc.ac.uk/Databases/Anatomy/XML/TS11.xml> (last viewed Jul 2006).

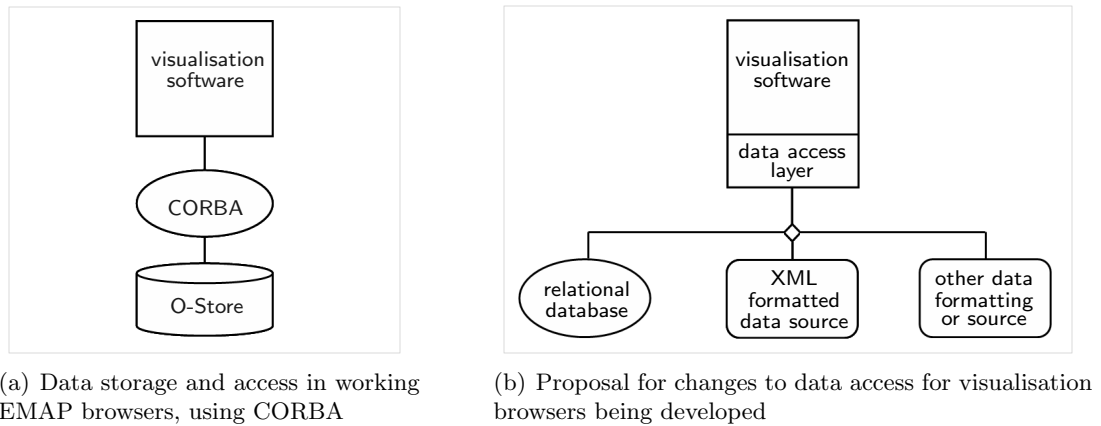


Figure 6.3. The working EMAP browsers read data from an OO database using CORBA and display the anatomy ontologies using indented text indices. The proposal for a solution to the limitations of the current browsers focuses on separation of the data from the visual analysis solution, to promote reusability and ease extension of the application developed.

to store each **AnatomyComponent** described, and generate an ontology tree. The **Loader** serves as the data access layer between the visualisations generated and the input data source and/or format, and is important to ensure extensibility and reusability of the application developed. Each *loader* class or method must:

1. determine the ontology type being loaded — current examples are:
 - a **DevelopmentStage** in an organism
 - an **AbstractOrganism** containing all components that occur in all stages of development of an organism.
2. parse the data and retrieve and store detail for each **AnatomyComponent**.
3. build an **AnatomyOntology** object as a tree, from a root and extending to its leaves, using **Relationship** objects to store the links between components. (Note that a **DevelopmentStage** or an **AbstractOrganism** is a specialisation of an **AnatomyOntology**. The application developed may be extended by writing classes to specify additional properties of other ontology types identified.)

Figure 6.4 shows data flow into the application and through the parser to create **AnatomyOntology** objects.

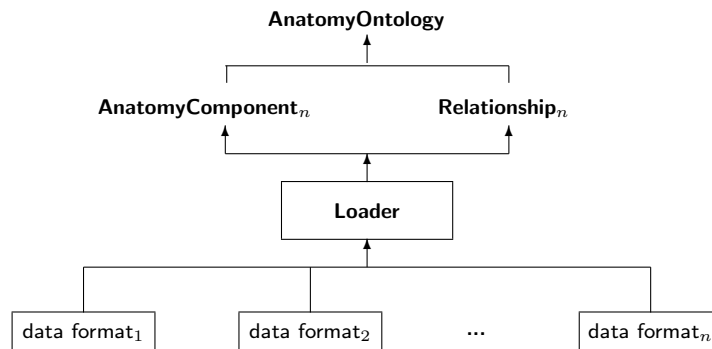


Figure 6.4. Input data flow for the visualisation browsers

6.3.3 The visualisation layer

Each **AnatomyOntology** created is used to generate and display an **AnatomyTree2D** or **AnatomyTree3D** as required, which is then displayed in the corresponding **2D-** or **3DTree-Browser**, as illustrated in figure 6.5.

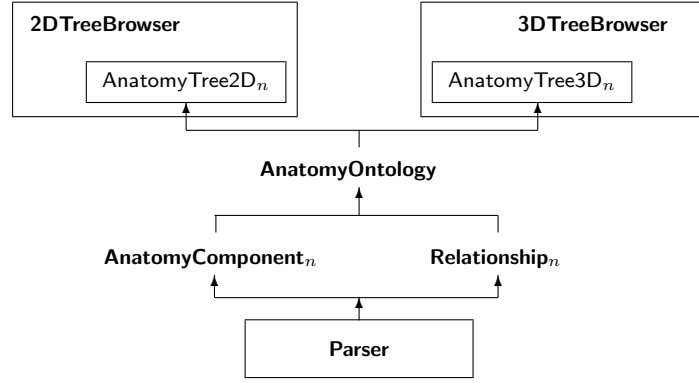


Figure 6.5. Structure of the visualisation layer

Both the 2D and 3D visualisations use rooted, node-link graphs to provide an overview of each ontology, with functionality implemented for detailed data analysis as discussed in § 6.5.4, § 6.7.5 and § 6.7.6. Figure 6.6 shows the overview for TS11, generated from the XML file for which an extract is shown in figure 6.2.

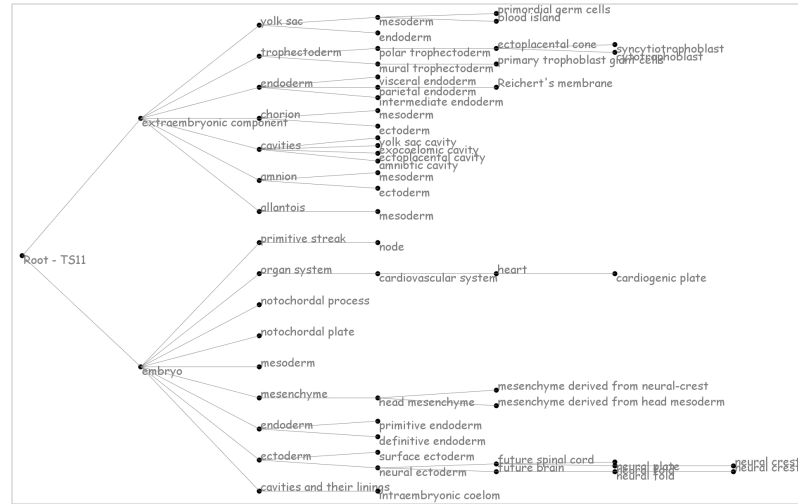


Figure 6.6. Graphical overview of the anatomy ontology for TS11 in 2D, using a rooted DAG with a horizontal orientation

6.4 Practical considerations in layout of node-link graphs

The merits and limitations of node-link graphs have been discussed in detail in chapter 3. Existing literature contains evaluations of different algorithms used to lay out data to make optimal use of screen space, descriptions of experiments performed to assess different tech-

because the distortion and constant change in structure associated with hyperbolic layouts would make it difficult to maintain a stable mental model of the structure of the data under study.

A radial layout was implemented at a later stage, to improve use of screen space. This was also to provide an interface familiar to a large portion of target users; the heuristic evaluation described in § 6.5.3 revealed strong support for the use of radial graphs, which are commonly used in visualisation of genomic data. Figure 6.8 shows the corresponding radial layout for the ontology data in figures 6.6 and 6.7.

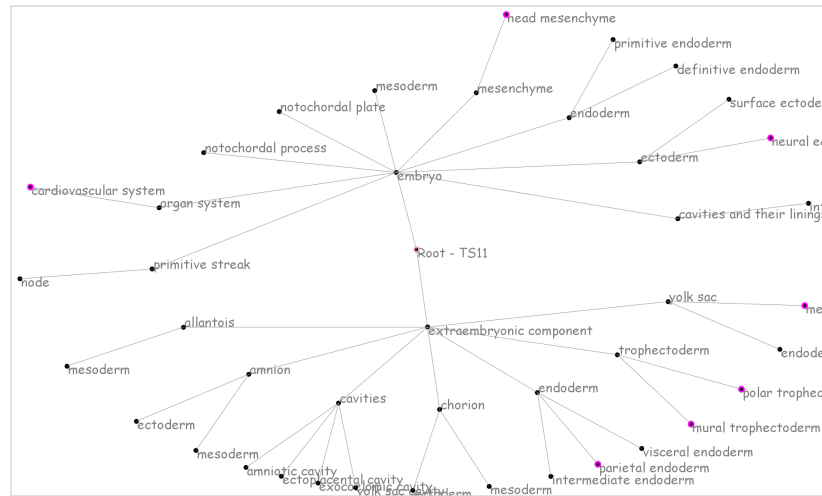
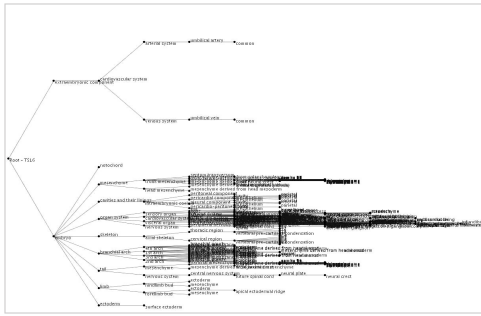


Figure 6.8. Radial layout for TS11, showing the first four levels in the DAG

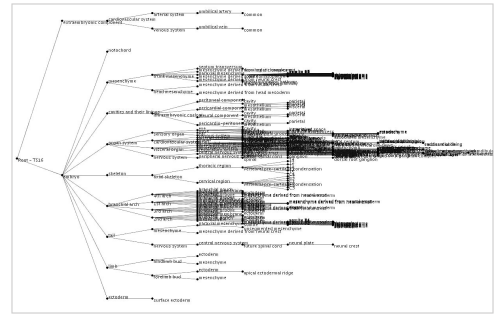
Merits of a relative layout

Two of the layouts described in [69] for the data classification tool, SimVis, are a *uniform* layout that distributes space equally among sub-trees, and a *relative* layout that distributes space to sub-trees dependent on the number of nodes they cluster. [69] found that the *relative* layout provided a better picture of clustering in the tree. This finding supports the views of users during the heuristic evaluation carried out: the initial layout of the DAG distributed space to nodes to obtain a *uniform* layout. This, however, made poor use of space available for drawing (especially large) non-uniform trees, as figure 6.9(a) shows.

Suggestions made by target users were to draw the graph to provide equal spacing to leaves, and lay out parent nodes successively up the graph toward the root. Irregular distribution of nodes in each graph, however, makes this a non-ideal solution. An improved layout, shown in figure 6.9(b), distributes space for drawing nodes in the layer immediately below the root, where the greatest bias occurs in node distribution, based on number of immediate children. Beyond this level, the layout returns to *uniform* distribution of space to sub-trees.



(a) *Uniform* distribution of nodes in the DAG drawn for TS16



(b) A *relative* layout weighted by number of children each sub-component of the root has.

Figure 6.9. Weighting space to draw sub-trees based on node distribution results in significantly better use of screen space. Data structure is also easier to discern in the *relative* layout.

Using abstraction to manage occlusion

The structure of the data lends itself well to abstraction, which is useful for managing the occlusion that occurs beyond a fairly small threshold. Only the first three levels in the ontology (including the root) are drawn when the graph is first displayed (see figures 6.10 and 6.11). This provides enough information to especially domain experts to begin analysis of the ontology of interest. The number of levels displayed in the tree may be varied interactively to reveal further information in each DAG as required.

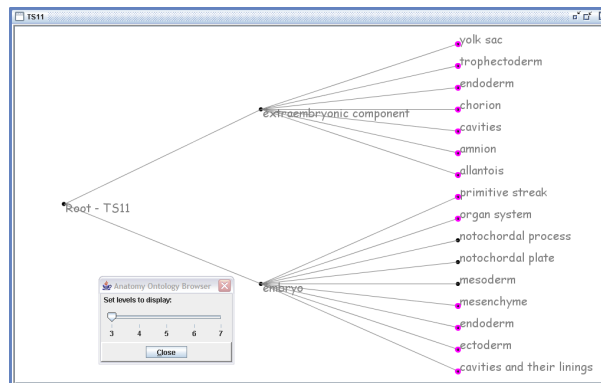


Figure 6.10. Compare the layout here for TS11 to that in figure 6.6, which shows all nodes in the graph. TS11 is a fairly small graph, containing only 61 nodes; for much larger graphs, such as for TS26 which contains 1749 nodes, shown in figure 6.11, abstraction has a significant impact on ease of especially interactive analysis.

6.5.2 Browser design

The Java application developed for visual analysis in 2D is shown in figure 6.12. The browser can display up to a maximum of ten internal frames simultaneously, each containing a single instance of the DAG drawn for the ontology of interest. The limit is imposed to manage memory required for the Java application to draw each graph, which increases with the number of nodes drawn to the screen, to maintain usability for interaction with the graphs.

An application menu provides access to all functions implemented. To increase usability

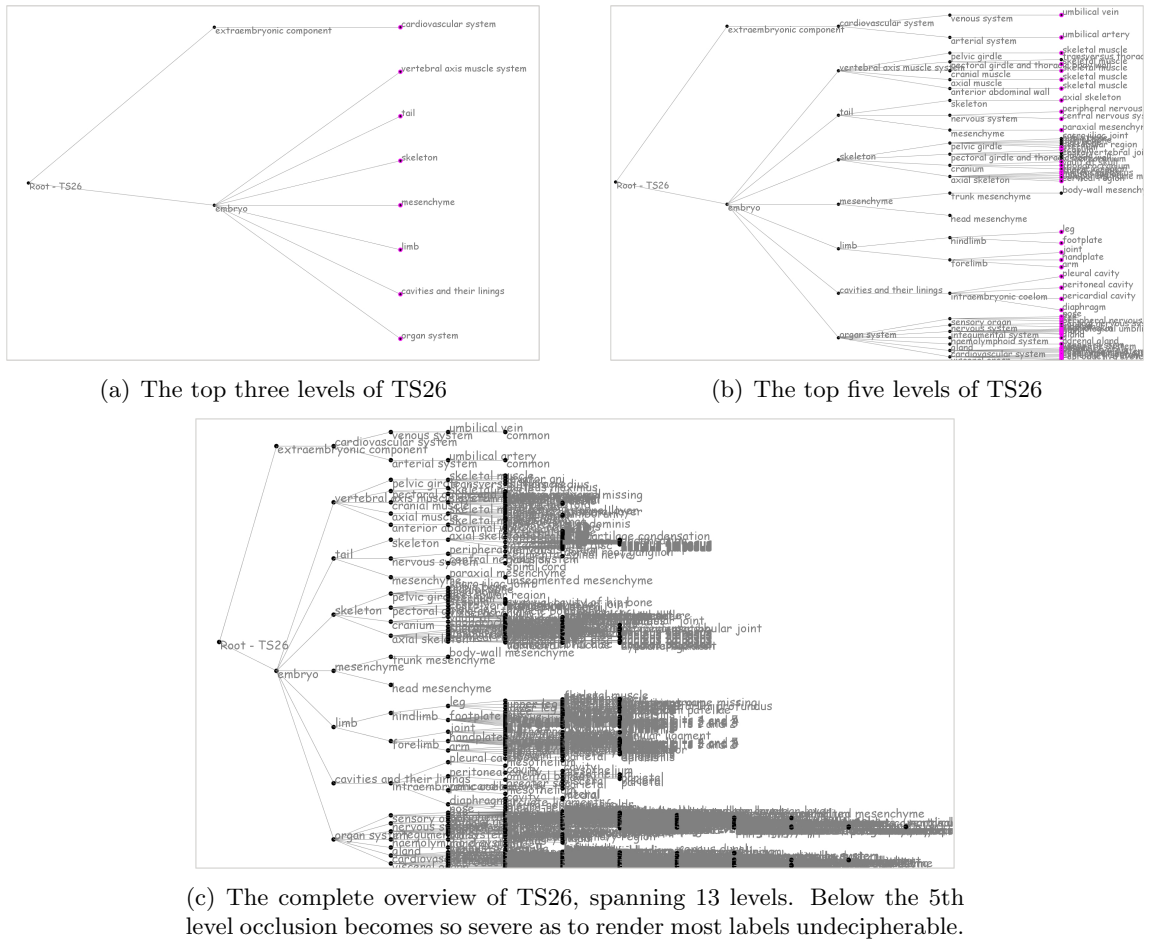


Figure 6.11. Abstraction used to improve usability of the overview graph drawn for TS26, which contains 1749 nodes.

for experienced use, shortcuts typically employed for accessing similar functions in GUIs are provided. A toolbar (whose structure is shown in figure 6.13) also provides quick access to the nine functions most likely to be used.

Four context (popup) menus are provided that match the application menu, but with functions grouped based on whether they apply to a single node, a selection of nodes, the entire graph, or a link between a node pair. § 6.5.4 details functionality available, and the structure of each menu can be found in Appendix B.

6.5.3 Heuristic evaluation of the initial prototype for the 2D browser

After a basic prototype had been developed as described in § 6.5.1 and § 6.5.2 (further information on functionality implemented follows in § 6.5.4), a heuristic evaluation was carried out to ensure that the application developed provided support for analysis as required by target users. The evaluation served a dual purpose: to ensure that design of the new tool would provide improved analysis of the ontology data being studied, and to prepare for a structured evaluation of the visualisation browsers being developed.

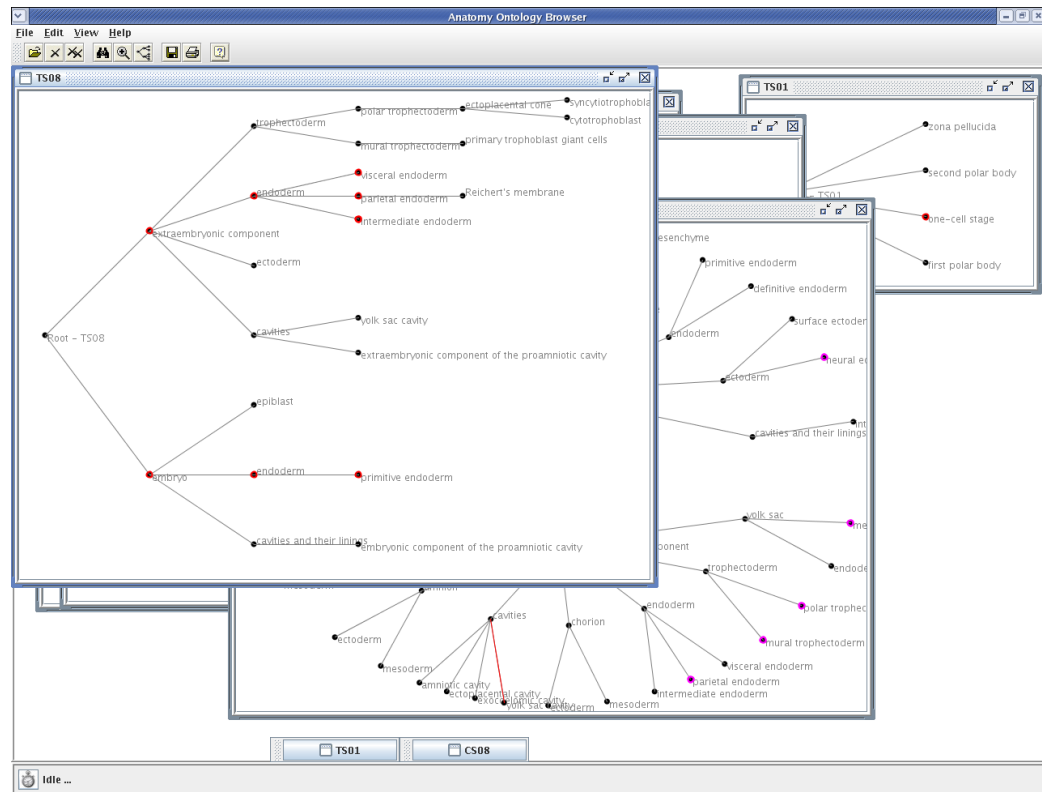


Figure 6.12. The visualisation application developed to hold the individual graphs drawn in 2D to represent anatomy ontologies.

-
- Open file / Load ontology
 - Close file / Unload ontology (with current focus)
 - Close all files / Unload all ontologies

 - Search
 - Zoom
 - Switch graph layout

 - Save system state to file
 - Print image
 - Help
-

Figure 6.13. Functions available from the toolbar in the 2D browser

Target user group

The evaluation involved researchers working at the Human Genetics Unit (HGU) of Edinburgh's Medical Research Council (MRC), on some aspect of EMAP and/or involved to some extent with XSPAN. The user group comprised biologists and computer scientists.

Procedure

The evaluation involved a demonstration of the functionality implemented for the 2D browser for interaction with and manipulation of the graphs drawn to represent the individual anatomy ontologies under study.

The (target) users were invited to comment on aspects of the system found to be useful for the analysis required, and functions that were unlikely to be used or would not make useful contributions to the research being done. Suggestions for improvements to the layout of the graphs and the functionality provided for analysis were also made.

Results

A major challenge in visualisation of especially large amounts of complex data is occlusion; an acute problem also encountered in the DAGs drawn for the system developed. Comments from users during the heuristic evaluation provided suggestions for alternative layouts, such as the radial layout commonly used to display phylogenetic and other trees in biology, to help overcome the occlusion that occurs. Further suggestions were to provide additional options for encoding data attributes such as changes to shapes of elements. Suggestions for improving the display of supplementary textual detail were also made. § 6.5.4 details options developed for analysis in 2D, highlighting suggestions made during the evaluation that led to changes and/or improvements in the prototype.

6.5.4 Options provided for analysis in 2D

Encoding of data attributes

The main method used for encoding data attributes is colour. The default colour of each node in 2D is black, with grey for links between nodes. Table 6.1 shows the colour coding used to distinguish attributes of different elements and graph structure.

Textual Detail

Print names (fully qualified name or path to root) were found to have the highest semantic meaning, especially to those users with prior domain knowledge (see also Table 7.1 and § 9.4.6). However, because *print names* grow successively longer as one approaches the leaves, labels default to (the shorter) *component names* for nodes. Labels may, however, be set to any of a node's properties, and the value for the node property currently set to display will be written to the graph provided the option to display labels is switched on.

Table 6.1. Data encoding in the 2D browser

Colour Code	Data Attribute
black fill	default node colour
light grey line	default link colour
magenta ring	collapsed node/hidden sub-tree
red fill/line	highlighted node/link
red ring	node with current focus/currently selected nodes
pale grey outline, no fill	ghosted node
green ring	search hit
orange ring/line	node/link along path drawn toward root
yellow ring/line	node/link along path drawn toward leaves
brown ring	node selected to form part of a new <i>group</i>
reddish-brown ring	parent node of a <i>group</i> node
yellowish-brown ring	child node of a <i>group</i> node

Complete detail for a node may be brought up by double-clicking on it, listing, where they are defined: (simple) *name*, *print name*, *IDs* of parent and child nodes, *abbreviations* or *synonyms*, the stage in which a component first appears, *start stage*, and that beyond which it ceases to exist, or develops into another component, the *stop stage*. Figure 6.14 brings up textual detail for the component *branchial arch* in TS12.

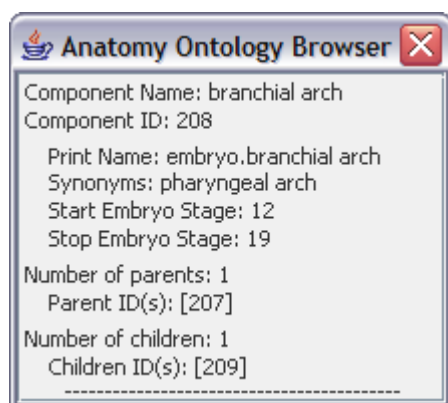


Figure 6.14. Component detail brought up for the node *embryo.branchial arch* in the graphical representation for TS12, using a custom dialog.

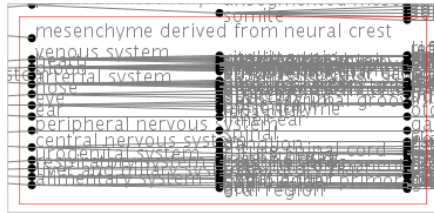
Hiding labels helps to manage the occlusion that occurs at a fairly low threshold, mainly due to node labels, and is especially useful where there is high density of nodes (compare figures 6.15(a) and 6.15(b)). In this case only nodes and links are drawn, and the label for the object with the focus is revealed as the mouse is moved over the graph.

Textual detail for links displays the default *part-of* relationship between nodes using the *print names* for the node pair of interest. The relationship between the nodes *embryo*, one of the two sub-parts of the root in TS12, and its sub-component *branchial arch* is therefore represented as *embryo.branchial arch* '*part-of*' *embryo*.

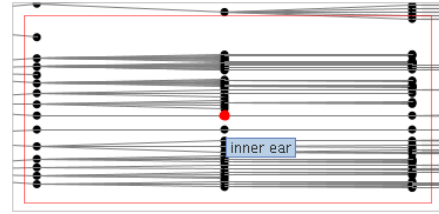
Highlighting and ghosting out data

Nodes of interest may be highlighted using a red fill, while ghosting may be used to suppress data of lower importance. Figure 6.15(a) illustrates a region in a graph with very high occlusion, both due to labels and overlapping nodes, in which a node of interest has been

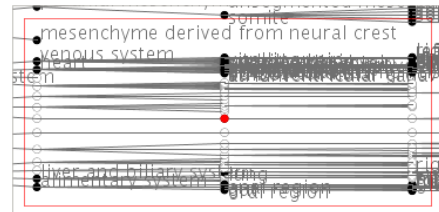
highlighted. In figure 6.15(b) labels in the graph have been hidden, resulting in a significant reduction in occlusion; the node is now easily identified as the *inner ear*. The label for this node is *popped up* when it receives the focus (achieved by holding the mouse over the node), also signified by the red ring drawn round the node.



(a) ROI in a DAG with a high level of occlusion that prevents nodes from being distinguished, and renders most labels illegible.



(b) One solution to occlusion — hiding labels in the graph. This *pops up* the label for each node as it receives the focus.



(c) Nodes surrounding the node of interest ghosted out in addition to hiding labels, to reduce occlusion in the ROI.

Figure 6.15. Ghosting out of nodes and hiding of labels to reduce occlusion in a graph, to allow focus on a single node of interest.

Ghosting initially only faded out the actual node, redrawing ghosted nodes with a pale grey outline, and so was not very effective. However combined with hiding node labels as shown in figure 6.15(c) it provided a degree of reduction of occlusion.

Selection of ROIs

There are two methods available for multiple selection of nodes: drawing a rectangular area to enclose all data of interest, and/or depressing the *Control-Key* while clicking on individual nodes. The latter, though more tedious, allows non-adjacent nodes to be selected. It should be noted, however, that if a region of interest is redrawn in a separate window this will only include nodes within the rectangular area drawn to the screen; redrawing non-adjacent and/or isolated nodes without the context of surrounding data reduces ability to understand relationships between the nodes of interest.

Selection of multiple nodes allows the same function(s) to be performed simultaneously on all data of interest, as figure 6.16 illustrates.

Expansion & collapsing of sub-trees

One method for obtaining abstraction of data is to fold away or hide sub-trees. This reduces complexity in the graph and also occlusion for areas with high density of nodes (and labels),

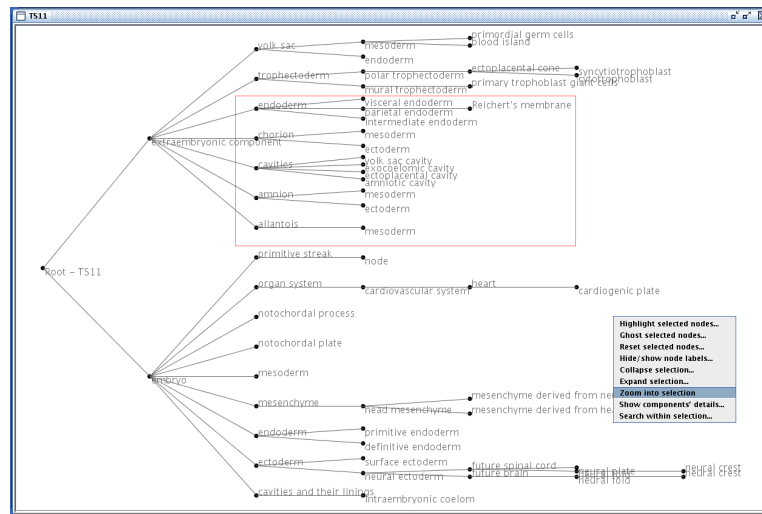


Figure 6.16. Selection of data of interest in the graph for TS11, allowing simultaneous editing of data properties for all nodes in ROI

providing more screen space for drawing data of interest.

A node whose sub-tree has been hidden is encircled by a bold magenta ring. Collapsed or *pruned* trees may be (re)expanded as required to reveal hidden data.

Zoom

The 2D browser provides four implementations of zoom. The first option allows users to redraw the rectangular selection area in a DAG in a separate, coupled window. This magnifies the ROI by redrawing nodes and links in a larger physical area, as illustrated in figure 6.17. The main disadvantage associated with this is the loss of the overview and hence, surrounding context. For small to average size monitors the sub-window, the **ZoomPane**, may overlap the main window, increasing difficulty in mapping between the detail window and the overview. For sufficiently large monitors this may be resolved by placing the windows next to each other, simplifying mapping between the two views.

The **ZoomPane** provides a limited set of functions for editing properties of nodes it contains, and changes made to nodes in the sub-window are reflected in the main window when the former is closed. § B.3 details the structure of the two context/popup menus provided for accessing functions in the **ZoomPane**.

A second option for zoom is to redraw only a sub-tree of interest in a separate window, providing both a physical and a semantic zoom, but again removing the context of the overview. Retaining context while examining detail was, however, seen by users to be an important requirement for effective analysis.

Two existing solutions to the loss of context, which results from drawing only a sub-set of data, involve the use of a hyperbolic or a magic lens. A hyperbolic layout would, however, result in constant distortion of the visualisation, with the danger that it might prevent a

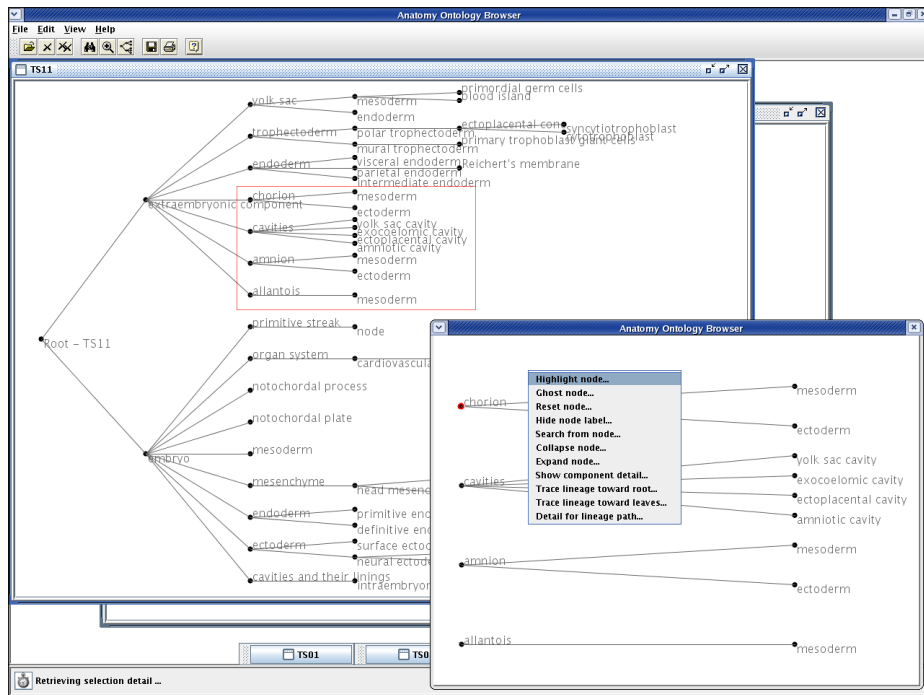


Figure 6.17. Nodes lying in the rectangular area drawn in the main window are redrawn in a coupled **ZoomPane**. The actual nodes are drawn at the same magnification, but the larger amount of space available means increased space between node pairs, eliminating or at least reducing overlap of nodes and labels.

stable mental model of data structure from being formed. A magic lens, making use of a uniform zoom, would exclude the distortion that is an artefact of hyperbolic layouts. However because it superimposes the virtual canvas containing the ROI at higher magnification on the main drawing area, it obscures data immediately surrounding the ROI, and so does not meet the requirements of the target user group.

A solution was finally developed that makes use of a hybrid between a hyperbolic layout and a magic lens, using a method similar to the zoom used in the SHriMP visualisation application [172]. The implementation developed for this project, however, zooms into a single sub-tree instead of magnifying individual nodes as is done in [172]. Figure 6.18 illustrates how sub-trees below the level containing a node of interest are folded away, allowing the sub-tree with the focus to be redrawn with a uniform zoom and using maximum screen space. Minimal rearrangement of the graph helps to maintain users' mental models of data structure, and also removes the extra cognitive load required for mapping between the coupled visualisations when the sub-tree of interest is drawn in a separate window. Further, context of surrounding data is maintained to a large degree.

The implementation is similar to that used in [139] who make use of a continuous zoom to provide smooth redrawing of the visual structure, constraining data drawn to the screen to the number of levels in sub-trees that can be drawn without limiting readability of text. User requirements for this project however differ from those in [139] in that ability to view the structure of a sub-tree of interest is considered to be more important than immediate availability of textual detail in less important ROIs. After obtaining the overview of the

larger sub-structure, additional functionality is provided for greater magnification and/or highlighting of the smaller ROI for those cases where users require further detail.

The last option is a geometric zoom which magnifies the entire tree in the main window, illustrated in figure 6.19(b). Magnification of the entire DAG was not implemented initially because the virtual canvas required to draw the magnified graph is larger than the physical space it is drawn in, so that it becomes necessary to scroll through the graph to navigate to ROIs. The context of the overview is lost, with the potential for an increase in disorientation during navigation. An advantage in this option, however, is that nodes are pushed further apart, reducing occlusion and increasing readability and hence, usability for ROIs. Users during the heuristic evaluation were of the opinion that the increase in usability, especially for regions of high density, would compensate for the loss of the overview.

Searching/querying

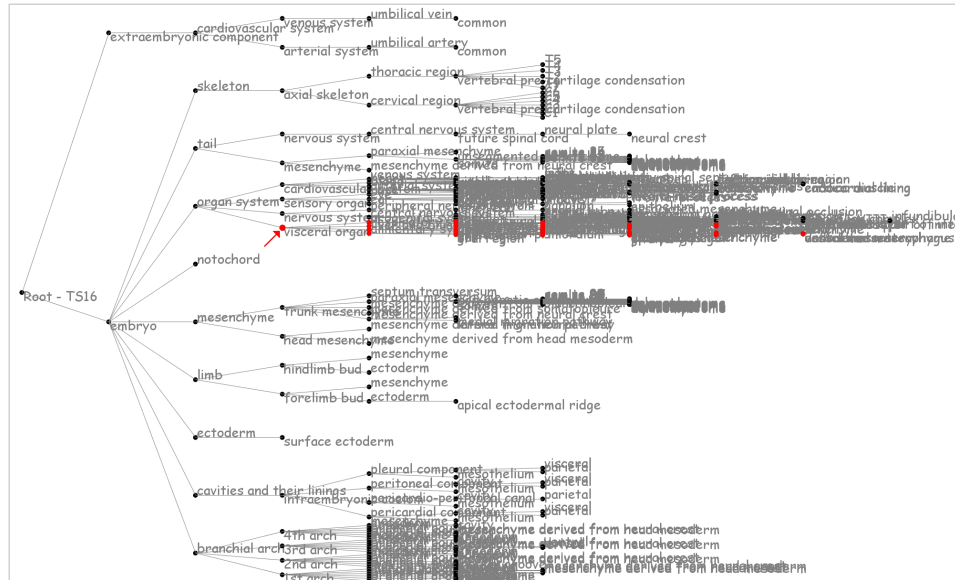
A custom dialog is used to perform sub-string searches on any of the properties defined for component nodes. Textual results are displayed in the search dialog, recording the number of hits and listing, for each match, *component ID* and *print name*. This is supported by graphical results that highlight corresponding matches in the DAG using a green ring round each node that satisfies search criteria, illustrated in figure 6.20. This makes it easier to recognise the distribution of search results and especially aids the identification of relationships between those objects with a large physical separation. Individual hits may be selected from within the search results, by double-clicking on entries for nodes of interest. This is especially useful for locating individual nodes where there is high density of nodes and/or a large number of hits.

Searching defaults to only visible nodes in the frame with the current focus, the only option initially made available. Users, however, found this restricting: options are now available for searching on all nodes in the current ontology, in which case a warning is displayed if any search hits are hidden. Alternatively searching may be confined to a sub-tree of interest or to nodes lying within a selection area drawn to the screen. A search may also be started from the node with the focus, in which case the search term defaults to the (value of the) *component name* of the node selected, and property to search on is set to *component name*.

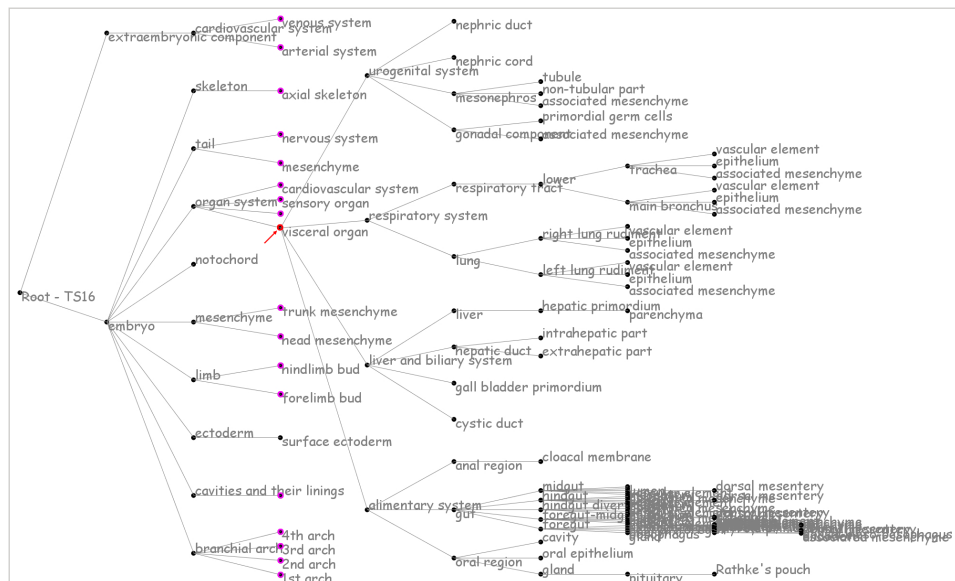
The system currently searches only within the ontology with the focus. In response to suggestions made during the heuristic evaluation, extension of the system will look at transparent translation of queries, to retrieve additional, supporting information from previously verified, relevant data sources.

Tracing (lineage) paths within an ontology

Visualising the anatomy ontologies under study provides an intuitive method for determining component parts of a node and identifying the (lineage) paths they follow in a DAG.

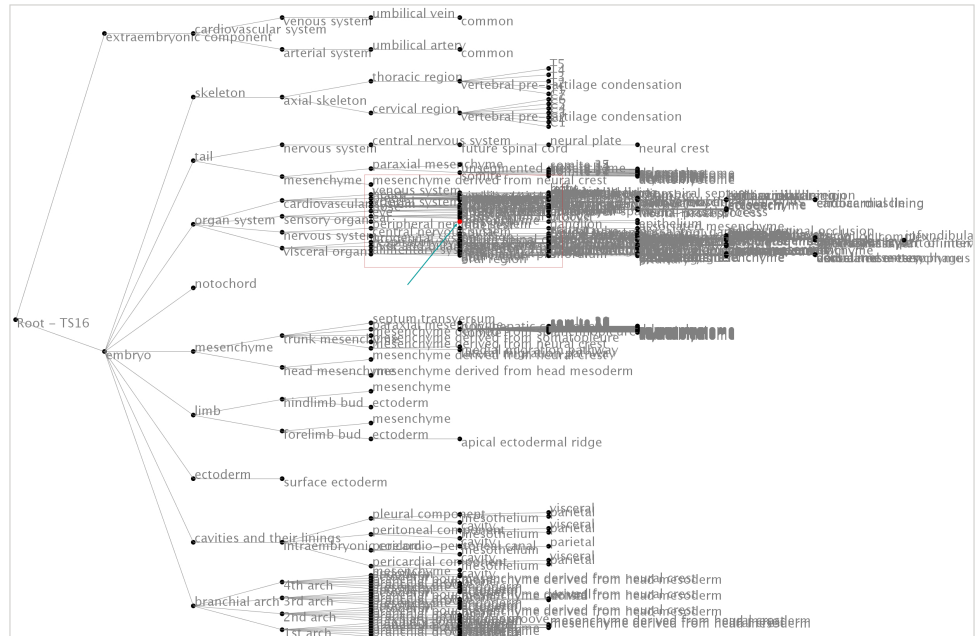


(a) The overview for TS16 at default magnification, highlighting the *visceral organ* and all nodes that make up its sub-tree.

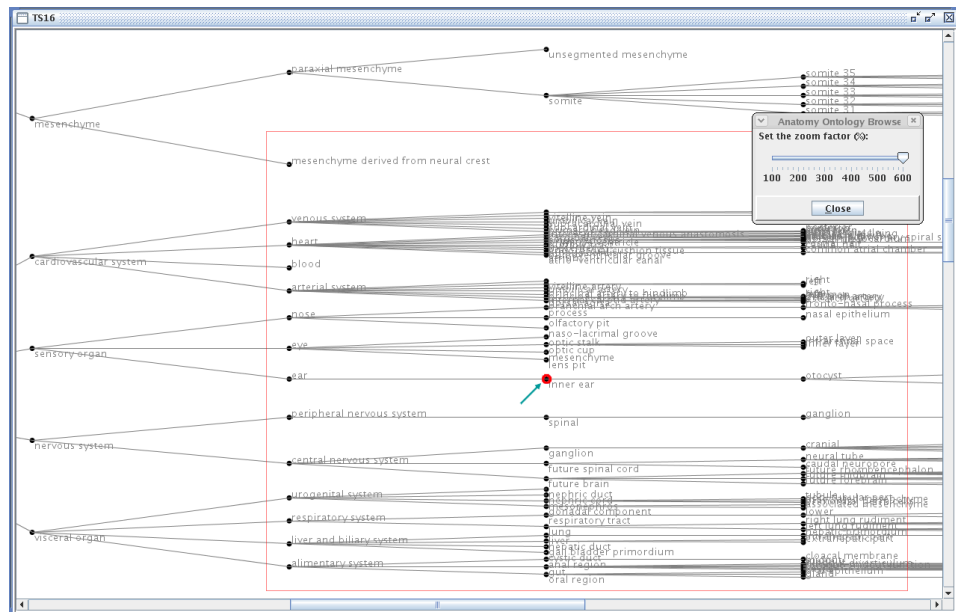


(b) The sub-tree for the node *visceral organ* (highlighted with a red ring) is expanded using maximum screen space, providing both a semantic and a physical zoom of its sub-tree.

Figure 6.18. Comparing fig 6.18(a) to fig 6.18(b) it is immediately obvious that magnification of the sub-tree of interest significantly increases the ability to analyse the ROI. The context of the overview is maintained, while the component parts of the node of interest are more clearly revealed, providing a semantic zoom that complements the physical zoom obtained in fig 6.18(b).



(a) A rectangular area is drawn in red to enclose the node of interest, the *inner ear*. (This is the same region shown in figure 6.15).



(b) Physical magnification of the ROI in TS16, to reduce occlusion in the graph

Figure 6.19. The region surrounding the node of interest, the *inner ear* (highlighted in red), is shown at maximum magnification - 6 times the default canvas size. There is a significant decrease in occlusion for the densely populated area shown at default size in figure 6.19(a).

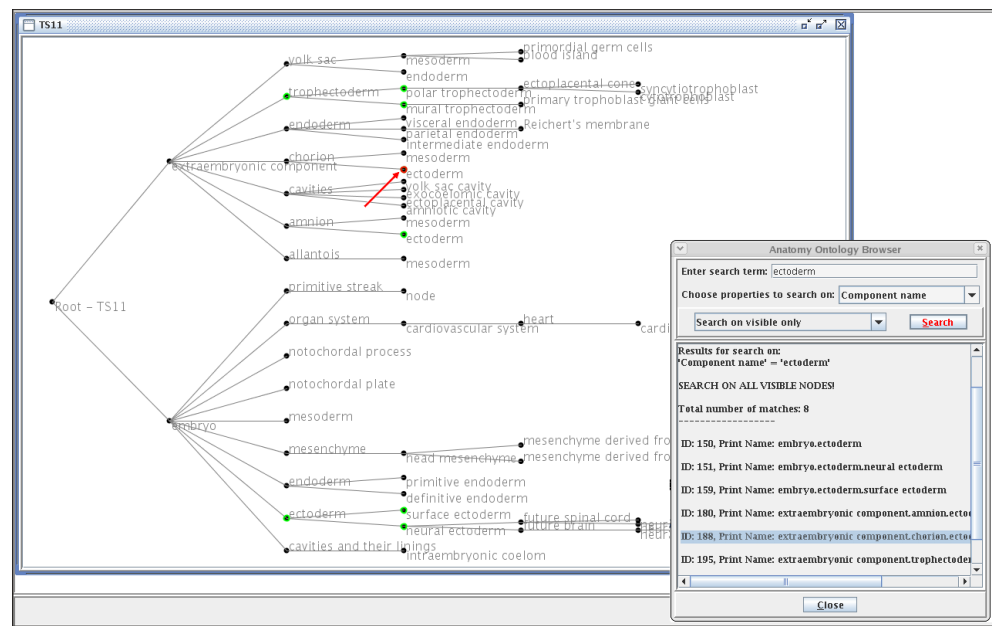


Figure 6.20. Data nodes in the graph satisfying search criteria are highlighted in green, and corresponding textual results are listed in the search dialog. The arrow points to the search hit selected from within the dialog.

Tracing paths in a graph toward the root successively reveals the components for which a specified component forms a part, highlighting nodes and links along a unique path (in orange), for a specified number of levels. Conversely, paths toward the leaves of a tree may be traced, successively highlighting all sub-components of a node and the links between them (in yellow). Figure 6.21 shows paths traced through TS11 from three components. Supplementary textual detail describing all nodes lying along a path may be brought up as required.

Creating alternative sub-structures

The need for *grouping* of nodes has been previously discussed (refer § 5.3.2 and § 5.3.4). The visualisation browser developed provides graphical support for creating *groups*, using a custom dialog to input values for properties of the *group* node to be created. Initial implementation allowed users to select nodes to form part of a group from a list. Users during the heuristic evaluation requested additional options that would allow clicking in the graph to select nodes or the ability to enter IDs directly in the dialog provided.

Two issues associated with the creation of *groups* is the crossing of links that may occur, as in figure 6.22, and a potential increase in occlusion due to the additional nodes and links drawn.

Resetting graph to default state

Physical editing of a node, a selection of nodes or an entire graph may be reset to the default state. Note that this does not remove annotation to nodes or links.



Figure 6.21. Three paths are traced within a single ontology, two highlighting the ancestors of the nodes *amniotic cavity* and *head mesenchyme* (in orange), and the third successively highlighting the components which form part of the node *neural ectoderm* (in yellow).

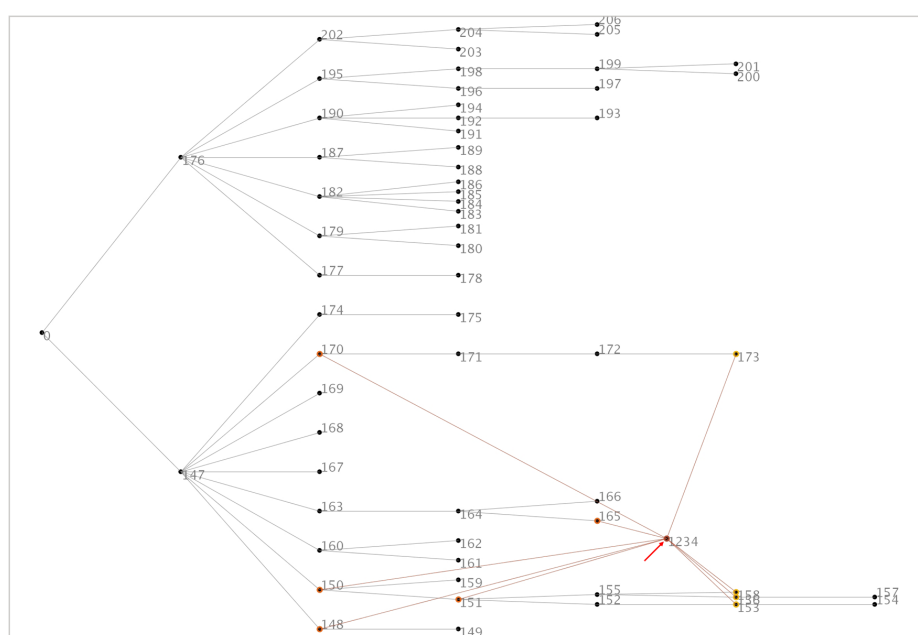


Figure 6.22. The horizontal layout is shown for a *group* created in TS11. The graph is no longer a true tree but a DAG, and some nodes now trace multiple paths to the root.

6.5.5 Limitations in the 2D browser

Two main challenges recognised in the 2D browser are occlusion largely due to poor use of screen space, typical of tree graphs, and exponential increase in system response time with data load. The following sections describe in more detail specific problems for which solutions were required in order to develop a usable system that also provides novel, intuitive visual analysis solutions for researchers.

Occlusion in DAGs

Beyond a threshold of about 200 nodes in a DAG occlusion begins to degrade visual analysis. The main problem is due to overlapping labels, and in areas of especially high density, overlapping nodes that make it difficult to distinguish individual components as occurs in figure 6.19(a). This is a problem common to node-links graphs, and is largely due to poor use of screen space (§ 6.5.1 contains a discussion on the use of different layouts to optimise use of screen space).

Simultaneous visual analysis of multiple trees

One requirement for the research being performed is the ability to compare multiple data sets simultaneously. This is to provide more intuitive methods for tracing lineage (refer § 5.3.3), and also to determine equivalence in components across ontologies. The information retrieved can then be used to infer structure and function of newly discovered genes in different organisms, by mapping to existing knowledge about gene expression data in other (related) organisms.

[130], among others, recognise that current hierarchical visualisation solutions are geared toward navigating through data, with limited support for comparing different data sets. Their solution to this problem was to develop the *TreeJuxtaposer* application, to aid biologists in the identification of equivalent elements across phylogenetic trees. This problem was confirmed by attempts to make use of traditional node-link graphs to perform the analysis required; a major challenge encountered is the limited screen space that makes comparison of multiple ontologies in 2D a non-viable option.

A number of options have been explored, seeking solutions to the problem of occlusion that occurs in the 2D browser. These include focus on ROIs in isolation, and highlighting selected data while suppressing surrounding data of lower interest. These solutions are however limited to analysis of sub-sets in a single ontology, still leaving unresolved the more complex problem of simultaneous visual analysis of multiple ontologies.

6.6 Visualisation in 3D? Resolving limitations in 2D

Extending the 2D visualisations to make use of the third dimension resolves the problem of insufficient space for displaying data, taking advantage of the larger amount of space

available in 3D. [151] found that the increased density of data in 3D visualisations, resulting from the storage of a larger number of objects using the same amount of screen space, enables more knowledge to be retrieved. [69] also observed that increased data density in 3D not only makes more optimal use of space but also leads to the formation of structures that provide cues that help to uncover useful knowledge stored in data. Further, additional cues provided by natural perspective in 3D prevent the increase in cognitive load that would normally occur with increase in data size [150].

The merits and limitations of 3D have been discussed in § 2.6; the decision to use 3D as a solution to the problems encountered in the 2D visualisations balanced the need for more space in which to display data against recognised difficulty in the use of 3D for visual, interactive analysis of complex data. Development of a 3D browser would allow multiple trees to be drawn, each representing a single ontology, using the same amount of (physical) screen space, but within a larger virtual space, allowing simultaneous comparison of multiple data sets.

It was decided to continue to use the relatively simple 2D layouts to generate overviews of individual ontologies, with the additional functionality developed for detailed visual analysis of ROIs. However for comparison of multiple anatomy ontologies the visualisation system would be extended to make use of the extra space provided by the third dimension, to allow the larger amount of data to be displayed in a single window. The extra space in 3D was expected to lead to improved analysis of the ontology data, by providing intuitive tracing of lineage across multiple stages of development in an organism, identification of equivalence in components across ontologies and in the grouping or classification of data elements to reveal alternative structures for presenting the data. Natural perspective in 3D would also provide some of the benefits of a wide angle lens without the distortion associated with a hyperbolic layout. Navigation through and exploration of the larger data sets were also expected to improve with the larger number of degrees of freedom available for use in 3D worlds.

6.7 The 3D browser

6.7.1 Choice of programming language

Arguments for building the new visual analysis system using Java were presented in § 6.1. For the same reasons: to provide an interface similar to those for existing EMAP tools, to develop an application that is cross-platform compatible, and to increase accessibility for both standalone and possibly, web use, Java3D seemed the best choice for development of the extension to the 2D browser described.

6.7.2 Design of visualisation graphs

Bearing in mind difficulty associated with the use of 3D visualisations, especially in environments geared toward the use of 2D displays, simplicity was an important factor in the design of the 3D graphs. Building on learning from the 2D browser this application continues to draw node-link graphs to represent the anatomy ontologies being studied, with each graph lying in a 2D plane. (3D) spheres are used to represent components, and 2D lines link component pairs through their centres.

The 3D browser develops an alternative to existing visualisation solutions, building a visual representation of the data that lays out multiple graphs in equidistant, parallel planes arranged along the horizontal axis, employing a layout similar to the use of 3D parallel coordinates and the *Cube* system described in [69]. The root node of each graph lies on a common plane parallel to the horizontal axis, and graphs grow downwards, with fixed separation between levels in each graph. The layout allows identification of distinct, individual anatomies, while still providing intuitive analysis and comparison of multiple data sets, as figure 6.23 shows. Users are able to control the degree to which they become immersed in the data, navigating through the data objects drawn or moving the camera above or below the data sets as in figure 6.24, to obtain an overview of all data and relationships between different ontologies.

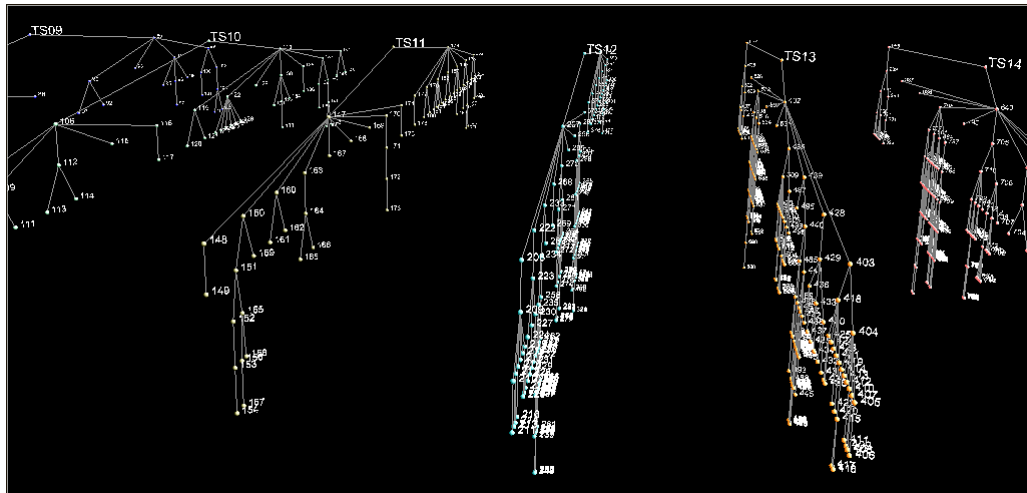


Figure 6.23. Six DAGs are loaded into the 3D window. DAGs lie in independent 2D planes arranged in parallel along the horizontal axis. Natural 3D perspective can be seen to increase magnification as data elements approach the viewpoint.

Figure 6.24 shows how the space between DAGs can be used to draw links between components in different ontologies, using colour to encode different types of relationships between node pairs. Relationships that cross data sets are easily identified as they stand out from the individual DAGs drawn.

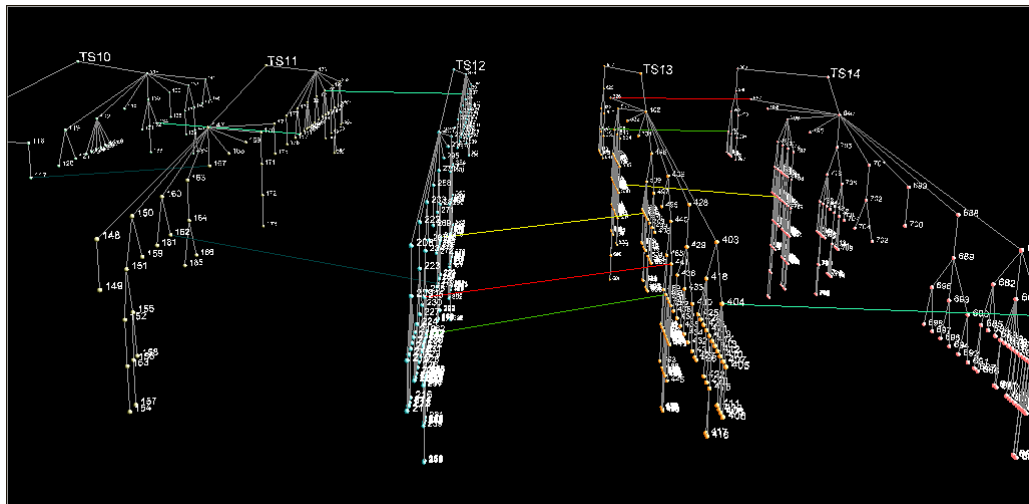


Figure 6.24. Moving the viewpoint away from the centre provides an overview of the DAGs drawn that highlights relationships between node pairs across different data sets. Colour coding is used to differentiate types of relationships occurring between nodes.

6.7.3 Browser design

All objects in the 3D universe are drawn to a single window allowing direct comparison between elements across data sets. As for the 2D browser a maximum of ten individual trees may be loaded at once. Memory management for the 3D browser is even more critical than for the 2D; implementation of a scenegraph that allows the interaction required for objects drawn in the 3D world prevents reuse of elements in the scene. Memory required to draw graphs therefore increases significantly with number of elements (nodes and labels) drawn to the screen.

Built-in functionality for navigation in the Java3D world includes zoom, rotation and translation or panning, all available using a three-button mouse. Alternatively the keyboard may be used for navigation. Tables B.6.1 and B.6.2 detail actions associated with **MouseBehavior** and **KeyNavigatorBehavior** Java3D objects and that are available for use in the 3D browser developed.

As for the 2D browser an application menu is provided, with shortcuts for functions expected to be called frequently. No context menus are available in the 3D browser because the right mouse button which is normally used to bring up popup menus is required for navigation in the 3D world. (The option to use the right mouse button with a function key was considered; however this results in an extra level of complexity and was therefore not used in the final implementation of the browser.) A toolbar is available, with functions as shown in figure 6.25.

Zoom makes use of in-built functionality in Java3D that moves in to the view, and combined with 3D perspective magnifies objects as they approach the viewpoint. The zoom button (found on the toolbar on the 2D browser) is replaced with a call to the dialog used to set colour codes for properties of elements drawn in the window. Only a vertical orientation of

-
- Open file(s) / Load ontology(ies)
 - Close file / Unload ontology (with current focus)
 - Close all files / Unload all ontologies
-
- Search
 - Set colour (data attribute) codes
 - Draw mappings between node pairs
-
- Save system state to file
 - Print image
 - Help
-

Figure 6.25. Functions available from the toolbar in the 3D browser

the graph is available for the 3D browser; the toolbar function in 2D for switching between different layouts is replaced by a call that brings up the custom dialog used to draw (external) links between components in different trees. Finally, the 3D browser allows multiple files to be loaded with a single call to *Open*. Appendix B details the structure of the application menus and the toolbar for both browsers.

6.7.4 Encoding of data properties

As in the 2D browser, colour is used to encode data attributes. To differentiate between individual data sets the application cycles through ten pre-set options for colouring nodes in each DAG. The colour assigned to each tree may however be reset as desired. *Part-of* links between nodes have a default encoding of grey. A legend, shown in figure 6.26, is provided that shows (editable) colour codes for all relationship types.

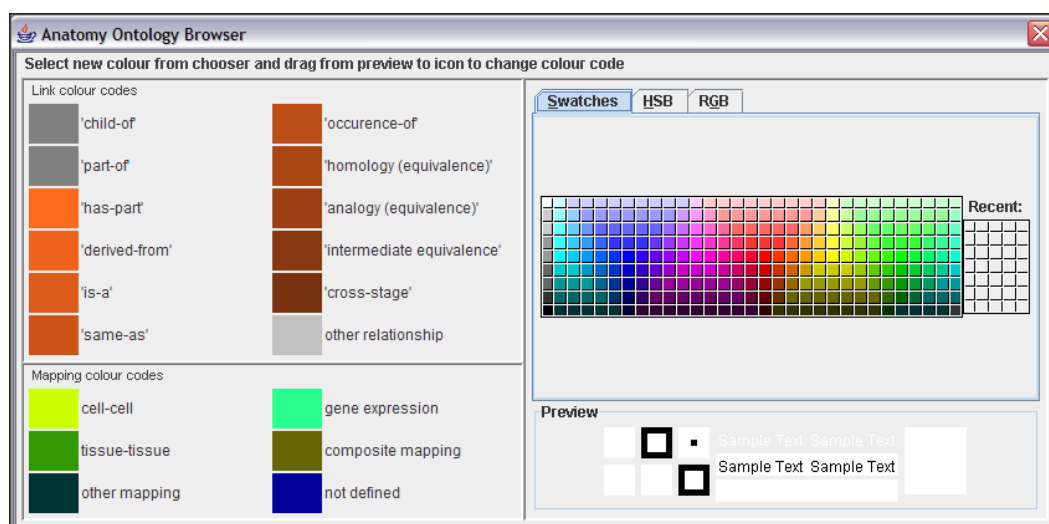


Figure 6.26. Editable legend displaying default values for colours used to encode attributes of objects in the scene

To maintain uniformity with the 2D browser the node(s) with the focus is/are highlighted in red. Search hits are coloured green and collapsed nodes filled with magenta (the 2D browser uses a ring for encoding search hits and collapsed or hidden sub-trees).

6.7.5 Overcoming limitations to analysis in 2D

Simultaneous visualisation of multiple ontologies

Because limited space in 2D prevents loading more than a single data set in a frame, simultaneous analysis of multiple graphs in the 2D browser results in significant cognitive memory load; users must map between elements lying in different, often overlapping frames. The additional dimension in 3D provides extra space so that it is possible to load multiple data sets in a single window, limited only by the memory required to draw objects to the scene; figure 6.23 shows six DAGs drawn in the 3D window. The following sections illustrate how the 3D browser simplifies tracing lineage in an organism and identification of similarity in function and structure of components in different organisms.

Tracing lineage during development of an organism

Lineage for a component identifies the stage of development in which the component first appears, and maps its persistence through subsequent stages till it ceases to exist or develops into another component. Figure 5.9 and § 5.3.3 describe functionality available for tracing lineage in EMAP, a method which places a large cognitive burden on users. A major requirement for the visualisation solutions being developed is to provide graphical support that eases tracing of lineage during development of an organism.

A first attempt to provide graphical support for drawing lineage paths required users to identify successive node pairs through which a path would be traced in order to draw links between each pair, across the space between individual DAGs. Figure 6.27 shows links drawn across DAGs to trace lineage, identifying persistence of component across stages, or components and/or their parts a specified component evolves into.

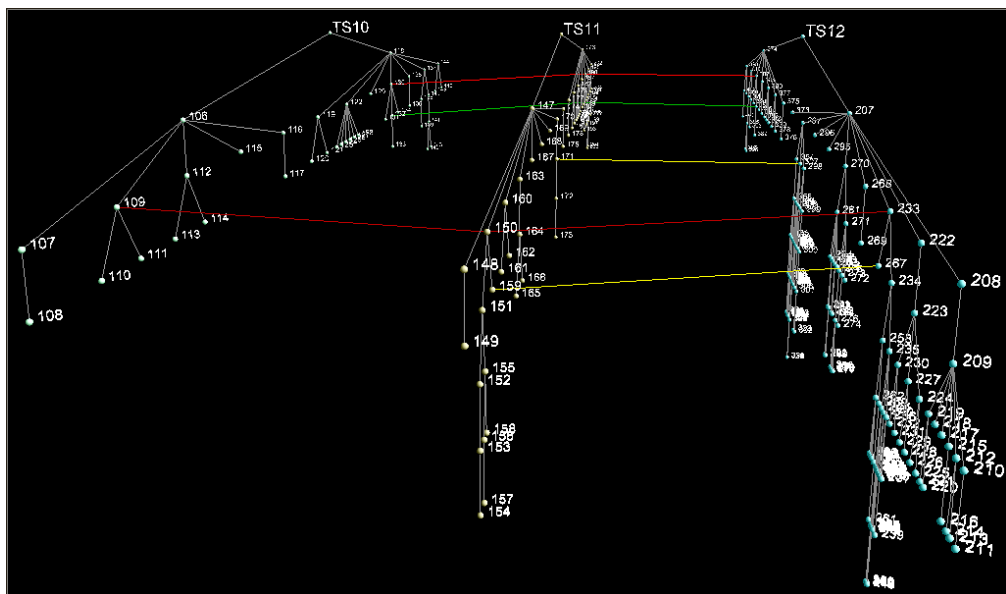


Figure 6.27. Colour-coded links drawn across the space between individual DAGs, to map lineage across successive stages of development.

The solution developed is similar to an initial prototype built by [91] while developing a visualisation system for comparing the results of different methods for classifying taxonomic data, to capture tasks performed by users. [91] draw hierarchical node-link graphs that encode similarity between elements using shape and colour of nodes, combining fading of objects of lower interest and highlighting of ROIs to aid analysis. Paths may also be traced through nodes in hierarchies drawn in sequence in a 2D plane, to highlight changes in classification of data. The prototypes drawn use very small graphs (two or three levels with less than ten objects drawn at each node); [91] recognise the limitations that 2D places on the amount of data that can be drawn to a display. The prototype is not developed further because user evaluations found an alternative using a set-based visualisation technique provided more effective representations of target users' mental models of data and analysis.

Mapping equivalence across multiple ontologies

Identifying components with similar characteristics is important in determining structure and function of newly discovered gene expression data. The custom dialog shown in figure 6.28 is used to record relationships between components that are not explicitly defined in the ontologies, but are identified based on expert opinion or evidence obtained from analysis of gene expression or other relevant data. The information recorded may then be used to draw links between components as in figure 6.30.

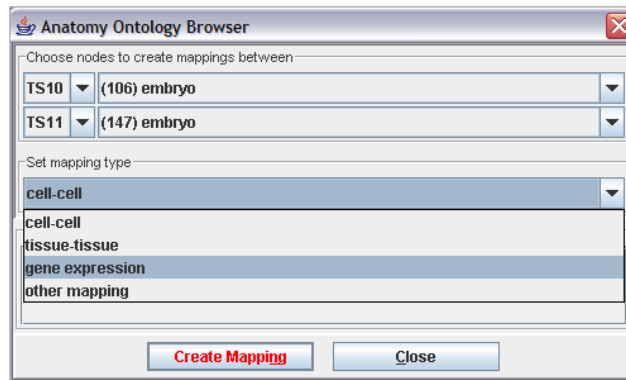


Figure 6.28. Custom dialog used to record different types of equivalence (relationships) between component pairs, to allow physical links to be drawn between DAGs in the 3D browser.

Grouping of data

The 2D browser provides graphical support for creating *groups*. However the additional links required to create the new *groups* may lead to increased occlusion or result in crossing of links in the 2D graph, as occurs in figure 6.22. An advantage obtained creating *groups* in 3D is that the *group* node created can be placed in a plane parallel to that holding the DAG it belongs to, simultaneously highlighting the *group* and eliminating the crossing of links that occurs in 2D, illustrated in figure 6.29.

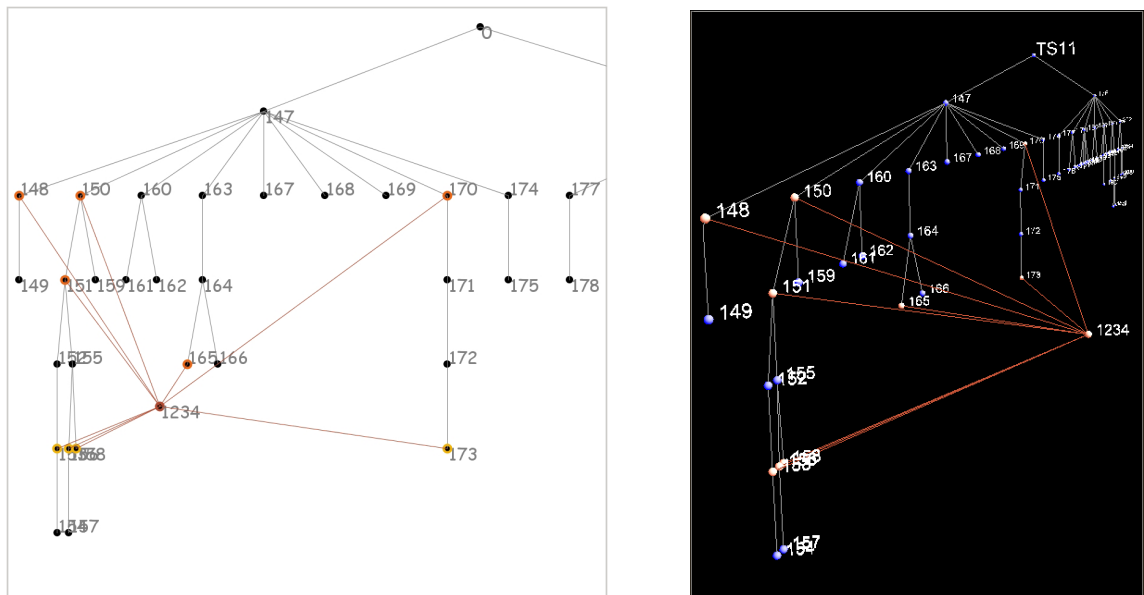


Figure 6.29. Grouping in 3D, shown on the right, is able to take advantage of the extra dimension to remove the *group* node to a plane parallel to that holding the tree it belongs to. This highlights the *group* created and removes the crossing of links that occurs for the equivalent 2D graph in the snapshot on the left.

6.7.6 Further options for analysis in 3D

Textual Detail

Double-clicking on any node in the graph or selecting a node(s) and choosing the appropriate item from the *View* menu brings up textual detail as for the 2D browser (refer figure 6.14). To minimise resources required to draw objects additional textual detail is not available for *part-of* links between nodes in the same ontology. Supplementary text describing (user-created) links across ontologies may however be displayed, as illustrated in figure 6.30. Information displayed includes *component name* and *ID*, and the types of relationships that occur between each node pair.

Highlighting nodes

The node or link with the focus is highlighted in red, as is each in a selection of nodes and/or links. In the same way as is done for the 2D browser, functions called are applied to all selected nodes and/or links as applicable.

Expansion & collapsing of sub-trees

Sub-trees may be collapsed into parent nodes, as in the 2D browser. Collapsed nodes are highlighted using a magenta fill, and hidden sub-trees may be revealed as required.

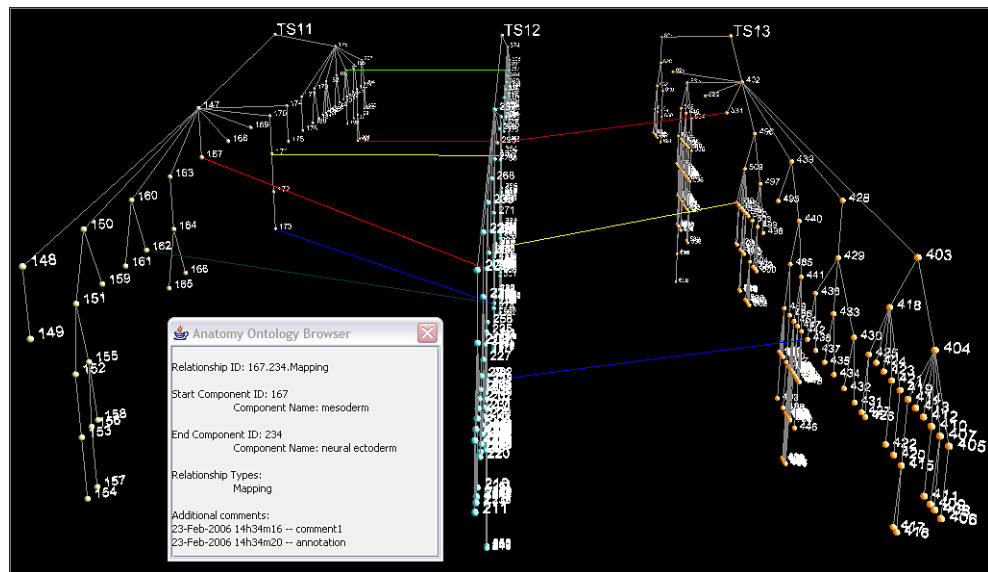


Figure 6.30. Types of relationships between node pairs are encoded using colour. Textual detail for a selected link (highlighted in red) describes the relationship between the two nodes it links.

Zoom

The 3D window makes use of in-built navigation options for zoom in Java3D (using the middle mouse button or the keyboard).

Searching/querying

Searching may be performed on any of the properties defined for nodes for all nodes drawn to the 3D window, with no options for restrictions on searching within data sub-sets. Figure 6.31 shows search hits highlighted in green.

6.7.7 Limitations in the 3D browser

The main limitation in the 3D browser is initial difficulty in navigation. The lack of a history function also means that users are not able to return to previous locations visited or move back to an earlier state. There is, however, in-built functionality that allows users to return to the default viewpoint — the centre of the universe, providing some measure of recovery if users become lost in the 3D world.

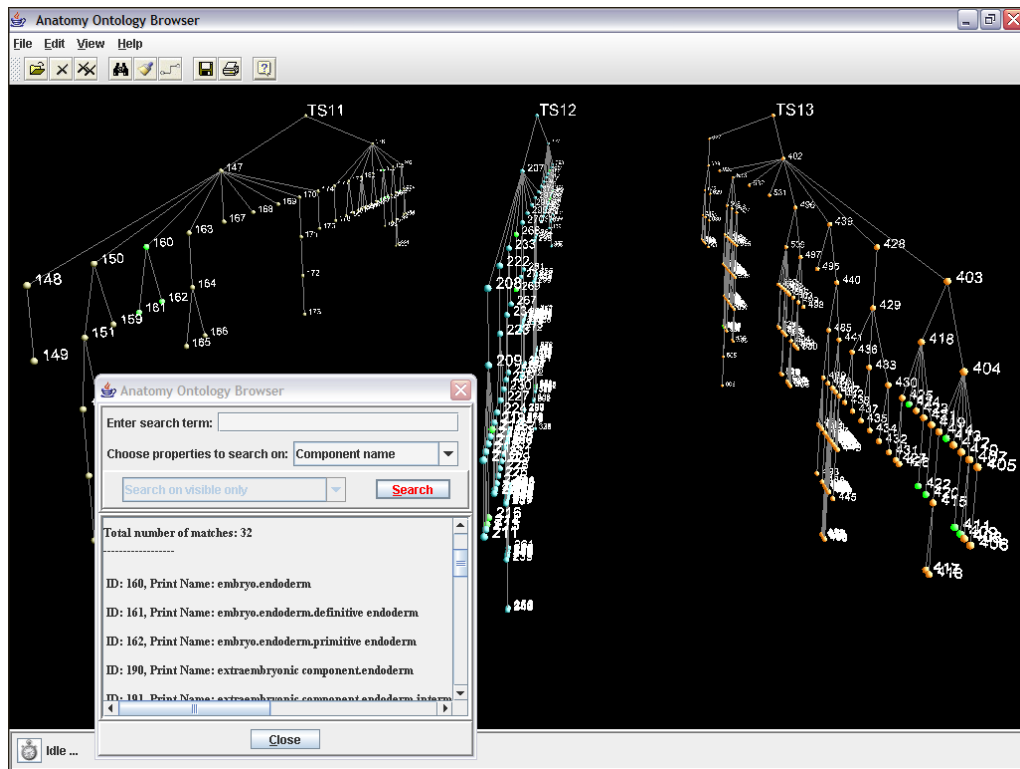


Figure 6.31. Nodes that match search criteria are highlighted in green in the 3D window

6.8 Related work in the field

Similar visual results to those developed for displaying relationships across data sets in the 3D browser [53, 52] (refer also § 6.7.5, and figures 6.27 and 6.30) have since been developed, illustrated in [102] and [103], using the same concept as that developed for this thesis: layering node-link graphs in parallel planes in 3D space, and drawing links between nodes across the space between graphs. [102] however build their visualisation as a 3D extension to the algorithm developed by [174] for drawing di-graphs, to visualise hierarchical data such as networks, DFDs and class structures in programming. They argue that improvements in resources for computing that provide better support for 3D visualisation can be harnessed to develop applications that make use of 3D to decrease complexity and increase understanding in data analysis. [102] split the data set used to generate a single 2D graph randomly to obtain two di-graphs lying on two separate *walls*. Figure 6.32 compares a 2D di-graph with the equivalent extension to 3D.

The comparison between the graphs drawn in 2D and 3D support findings in this thesis that show the advantages in the extra dimension (in 3D) for analysis of relationships in data; [102] redraw the different structures that make up the original 2D visualisation and highlight relationships within the data with greater clarity using the 3D representation of the same graph.

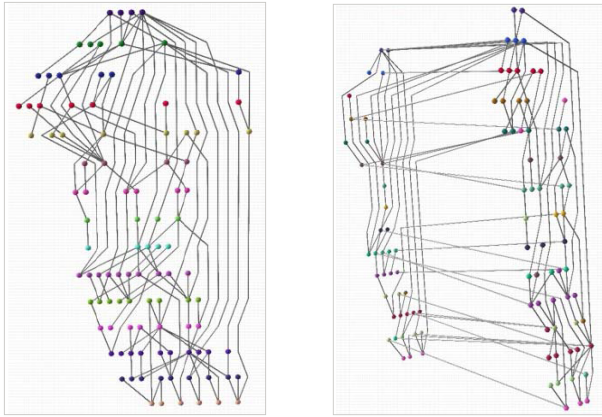


Figure 6.32. Layered di-graphs in 3D space that provide an extension to the technique described in [174] for drawing di-graphs. The 3D graph shows a reduction in edge crossings from 180 in the 2D graph to 24 in the 3D. Images reprinted with permission from [102]

Previous work has been done in the comparison of (independently created) ontologies, to determine overlap in knowledge stored in ontologies in related domains and aid data exchange and communication. Similarity may be determined based on lexical analysis — similarity in terminology used for naming concepts, and/or on structure — the relationships among data elements in each ontology. Limitations in these approaches include differences in interpretation of terms used, especially where different fields are involved in ontology creation and use; harnessing domain knowledge of experts is often required to ensure correct interpretation of terminology used to describe concepts.

One of the most well-known ontology alignment tools is the *PROMPT*² plug-in in Protégé, which uses regular and indented lists to display ontologies being analysed. *Chimaera*³ is another ontology editor that merges ontologies based on mappings identified between elements that describe data in similar or overlapping domains. *Chimaera* makes use of a forms interface for comparison of ontologies.

Though the forms of presentation in text-based systems such as *PROMPT* and *Chimaera* do not necessarily detract from analysis neither of these is able to present a complete overview of data structure. The advantage in a visual system such as that presented in this thesis is the ability to display (the structure of) a data set in its entirety in a single view, in addition to the more detailed analysis available in selected ROIs in both text-based and visual analysis systems.

Most text-based systems are limited to comparison of only two data sets at a time; among other considerations, cognitive load associated with textual analysis would increase difficulty tracing and tracking mappings across several data sets. Visualisation on the other hand allows easier identification of relationships within single data sets and those that cross multiple data sets; the visual system developed in this thesis allows different relationships to be viewed and compared more easily across multiple ontologies. This allows users to browse the data more easily to retrieve information, complementing (keyword-based) searching, the

²Information on PROMPT can be found at: <http://protege.stanford.edu/plugins/prompt/prompt.html> (last viewed Jul 2006)

³Information on Chimaera can be found at: <http://www.ksl.stanford.edu/software/chimaera> (last viewed Jul 2006)

latter of which requires domain knowledge.

6.9 Summary

This thesis has found that limitations to current data analysis are largely due to specialisation resulting in a restricted set of functions in individual data analysis and visualisation applications. Limited scope for integration between applications further increases difficulty exchanging data between tools, necessary for continuous or incremental analysis employing functionality provided by alternative tools.

This chapter described the development of a 2D visualisation application for anatomy ontology data, to provide analysis of individual data sets using first an overview, followed by the ability to analyse ROIs in detail, as suggested in the *“information-seeking mantra”* in [163]. The 2D browser provides a set of functions that may be used in concert to satisfy some of the information requirements of the test target user group (refer § 5.3), based on learning from tools developed to meet similar requirements.

The browser was also used as the basis for an evaluation of existing methods for visual data analysis; a heuristic evaluation was performed to determine if the solutions developed provide additional benefits to target users. Information obtained was fed into further development of the 2D prototype, and a second browser was built with a limited set of functions for visual analysis in 3D. This led to the preparation for a structured evaluation, to measure usability of the tools developed and compare options available for analysis in 2D and 3D. Further feedback was also required on the alternative options being developed to deliver improved visual analysis, to provide additional information that would lead to resolution of outstanding analysis problems. Chapter 7 details the usability evaluation performed for the visualisation browsers and analyses results obtained.

Chapter 7

Structured usability evaluation of visualisation prototypes

This chapter starts with a description of the preparation for the first structured usability evaluation performed for the prototypes developed. The actual evaluation process and results obtained are then presented, followed by a discussion of the findings. The chapter concludes with suggestions for changes and/or enhancements to functionality already implemented and for novel functionality for visual data analysis, based on research done within the scope of this project, user requirements and an assessment of functionality available for analysis in existing tools.

7.1 Preparation for usability evaluation

The aim of this evaluation was to obtain measures of user satisfaction with and effectiveness of the tools developed for the analysis required. The information obtained was used to determine the usability and utility of the applications developed; if a new tool was to be adopted by the target researchers it would have to provide advantages over applications in current use. This meant providing not only novel functionality with improved methods for data analysis, but also the ability to integrate new tools with existing ones, allowing at the least data exchange with applications already in common use.

Other issues the evaluation studied were the usability of the interfaces created and similarity to tools currently in use, to obtain a measure of learnability. Intuitiveness of navigation through the data especially for the 3D prototype, system response, and error handling were also evaluated.

7.1.1 Test hypotheses

Two sets of hypotheses were used to guide the evaluation process:

Null hypotheses

H_{0A} Textual analysis of especially large, complex data sets is as effective as visual analysis.

H_{0B} Visualisation in 3D provides no significant advantages for analysis over 2D.

Alternate hypotheses

H_{1A} Visual analysis of especially large, complex data sets provides significant advantages over textual analysis.

H_{1B} Visualisation in 3D provides advantages for analysis over 2D that justify the larger amount of support required (for development and use).

7.1.2 Preparation of evaluation documents

Task scenarios

A set of scenarios was developed to simulate target users performing tasks in their normal working environments. Successful completion of these tasks required (relevant) domain knowledge and an understanding of how functionality in the prototypes developed matches user needs. (Successful) completion criteria (SCC) were recorded and maximum time required to carry out each sub-task (MTC) estimated, to provide benchmarks for assessing users' responses.

A *walk-through* of the scenarios was performed by three evaluators, to ensure that they captured typical user tasks and allowed users to explore fully functionality provided in the visualisation prototypes. The evaluation team comprised a typical target user working at the MRC on the EMAP project, and two members of the XSPAN team.

Following this a trial run was performed, recording time to carry out individual tasks in addition to user reactions, to confirm that the tasks detailed could be successfully carried out.

Figure 7.1 shows an extract from the task sheet for the 2D browser (the complete task sheet can be found in § C.2).

The *walk-through* confirmed that the overviews provided (of the anatomy ontologies) did improve analysis. It also highlighted further functionality required to increase intuitiveness in analysis, to ensure that useful results would be obtained during the evaluation process. Changes were made where functionality as implemented would have a negative effect on usability and increase difficulty carrying out further evaluation, prior to performing the structured evaluation this chapter goes on to describe. Other functions were modified as required at later stages in the project. Table 7.1 details improvements suggested and those changes made prior to carrying out the structured evaluation.

Structured usability evaluation of visualisation prototypes

TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
1. Load Theiler Stage (TS) 11 in the browser.	REQ: 2D anatomy browser, Quick Guide SCC: Visualisation of TS11 displayed in browser MTC: 10s	N/A
2. Identify the anatomy component <i>chorion</i> and list the components which are 'part-of' <i>chorion</i> (immediate children of), as well as the Theiler Stages through which they persist.	REQ: 2D anatomy browser, Quick Guide SCC: Expansion of the DAG to show at least the component <i>chorion</i> . Components that are 'part-of' the <i>chorion</i> may be identified by tracing down the tree. An alternative is to bring up the component detail for <i>chorion</i> which lists child IDs. MTC: 60s	ID: 188 ectoderm Stgs 11-12 ID: 189 mesoderm Stgs 11-11

Figure 7.1. Extract from the task scenario sheet for the 2D browser for the structured user evaluation. The two columns highlighted in red detail completion criteria and MTC for each task. The complete task sheet can be found in § C.2.

Table 7.1. Proposals for changes to visualisation browsers prior to structured evaluation

Original implementation	Suggestion(s) for improvement	Implemented
Occlusion due to node labels: occurs even where occlusion of nodes is very low to none, especially for the TD layout	Hiding of labels (already implemented)	✓
	Interactive repositioning of nodes/labels	X
	Drawing labels at a (user-defined) angle to horizontal plane	X
Default labelling of nodes: set to component name	Change to print name (full path to root) to aid differentiation between nodes with identical component names, e.g., TS12 has four nodes with component name <i>mesenchyme</i> . (Note that this solution increases the problem of occlusion due to node labels.)	✓ (implemented in some cases)
Component detail: component IDs only provided in some cases	Provision of print names in addition to component IDs, as otherwise required to look up names to identify nodes	✓ partially implemented)
Search	Ability to highlight search hits in graphs from within search dialog	✓
Creation of groups: group nodes could only be selected from list held in dialog	Ability to click to select group nodes in the DAG of interest	✓
	Ability to enter component IDs directly	X
History/Undo function: not available	Undo function encourages exploration, especially for 3D where navigation sometimes produces unexpected, undesirable results	X
Storage of user sessions: not available	Provides history function	✓

Questionnaires

Two custom questionnaires were created: a pre-evaluation questionnaire to collect demographic information about users, and a post-evaluation questionnaire to record users' opinions about usability of the visualisation browsers. These were then reviewed by a Human-Computer Interaction (HCI) expert who is also a member of the XSPAN team. This was to ensure that the questionnaires would elicit useful information on usability of the prototypes, and ascertain that the functionality required was provided, and through an intuitive interface.

Initial suggestions for changes were on the wording, style and format, to make the questionnaires less terse. The layout of the responses also required editing, to provide optimum feedback. Modifications were made using the Questionnaire for User Interface Satisfaction (QUIS) [166] as a base. Each question was then examined and further changes were made to eliminate ambiguity in wording, bearing in mind that the two main user groups, with different research backgrounds, may interpret data and terms used in different ways.

The final structure of the pre-evaluation questionnaire provided options for users to choose from to provide information on gender, educational background and current field of work. Questions also addressed computer hardware and software normally used for work, experience with data analysis and visualisation applications, and prior use of the working EMAP section browsers.

The first part of the post-evaluation questionnaire gathered information on users' previous experience with computers. The main section presented closed questions with bi-polar answers on a Likert scale from 1-9, and the option N/A (not applicable) as required, covering the following topics:

- overall reaction to the system
- visualisation of data and suitability of the screen
- terminology used and system feedback
- learnability
- capability of the system.

The questionnaire ended by inviting users to comment on aspects of the application or the evaluation they felt had not been sufficiently addressed. A copy of the pre- and post-evaluation questionnaires can be found in appendix C.

A final suggestion was to make use of the SUS (System Usability Scale) questionnaire [27], to determine overall user opinion of usability using the well-tested questionnaire.

Other evaluation documents

To conform to ethical regulations in the evaluation process a user instruction sheet and consent form was also prepared (a copy of which can be found in § C.1).

7.1.3 Pilot test and expert review of evaluation procedure

A pilot test was carried out with an independent HCI expert with a specialisation in visualisation (a research student at the School of Maths and Computer Sciences, MACS, at Heriot-Watt University, HWU). The test was used to assess the design of the evaluation procedure, to ensure that the evaluation would elicit information from target users that would allow effective analysis and retrieve information on usability of the tools developed. A second aim of the pilot was to ensure that usability requirements had been built into the evaluation procedure.

Using an HCI expert for the pilot test also provided additional feedback on the structure and wording of the questionnaires. Main suggestions were for further changes to wording to remove ambiguity, and for improvements to the layout that would better distinguish different sections of the questionnaires and improve readability. An important problem in the post-evaluation questionnaire was also identified: a few questions required users to comment on functions not explicitly tested by the task scenarios. This posed the danger that users would report how they expected the system to work and not actual system response and functionality. The scenarios were edited to correct this.

A final review of the evaluation procedure was carried out based on the results of the pilot test, editing the evaluation documents as required. (Copies of all evaluation documents can be found in appendix C).

A timer and logger were then built into the visualisation browsers, to record each function called during the evaluation process. This was to supplement manual records of user actions to complete goals. An example of a typical log can be found in § E.6.

7.2 Assessment of variation in system response in 2D

A recognised and significant limitation of the 2D browser was the large increase in system response time with number of data nodes (and hence links and labels) drawn to the screen. To assess the severity of the problem two sets of tests were run late morning to early afternoon on separate days to minimise the influence of system and network load. The 2D application was run in each of Microsoft Windows XP[®], remote access of a Unix[®] terminal using Exceed[®] and directly in Linux[®]. Three computers were used to perform the tests:

- a PC running Windows XP with a P4 (Pentium 4), 2.2GHz processor, 256MB RAM (random access memory) and a 40GB hard drive

- remote access of a Linux[®] box with a P3, 933MHz processor, 256MB RAM and a 20GB hard drive
- direct run from a Linux terminal, running on a PC with a P4, 1.7GHz processor, 256MB RAM and a 20GB drive.

Monitor sizes were all 17in, with resolutions of 1024 * 768 pixels.

A single data set was loaded and unloaded, to ensure that all resources required to run the application were available and that all (global) application variables had been initialised before running each set of tests. The time lag in seconds to redraw the DAG for each of TS04 (11 nodes), TS11 (60), TS12 (198), TS18 (740) and TS26 (1748) from start-up showing only the top three levels to display all nodes was then recorded for six instances, disregarding the first load for each data set. The test results are summarised in figure 7.2. System response times for TS01 to TS11 are negligible, increasing slightly for TS12 with 198 nodes. Beyond this threshold there is significant increase in response time, with a corresponding negative impact on ease of interaction.

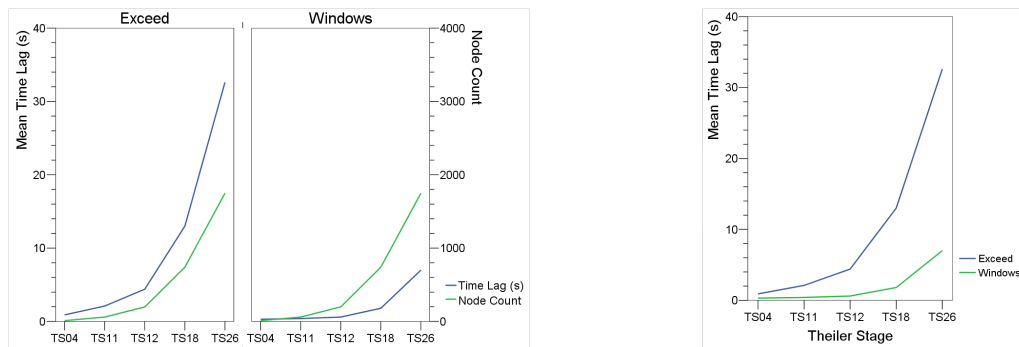


Figure 7.2. The two plots on the left show exponential increase in system response time with number of nodes drawn to the screen for MS Windows and Exceed. On the right a direct comparison between the two systems highlights significantly poorer response for Exceed.

The plots show even poorer response when running the program in Exceed. The difference in response is not just due to the faster system used in Windows; there is a noticeable decrease in program execution speed when running Java Swing programs using a remote terminal. (Results when running the program directly in Linux are similar to those in Windows and are not plotted separately).

Figure 7.3 shows the experiment repeated after the evaluation was carried out, when methods for laying out the data had been improved, averaged over the same remote system and a second Linux box also accessed remotely, with double the RAM and running on an AMD Athlon[®] processor at 2.4GHz speed. The experiment was also repeated in Windows, using a Pentium 4, 1.7GHz processor and 256MB RAM. Execution speed in Windows improved significantly, recording a maximum of 3s to redraw TS26, compared to a maximum of 8s for the faster system in the first experiment. There is however no improvement in Exceed for the same system. The faster remote system does record shorter system response times so that the overall average for Exceed is lower. However, times recorded are still significantly longer than for the system with lower resources in Windows.

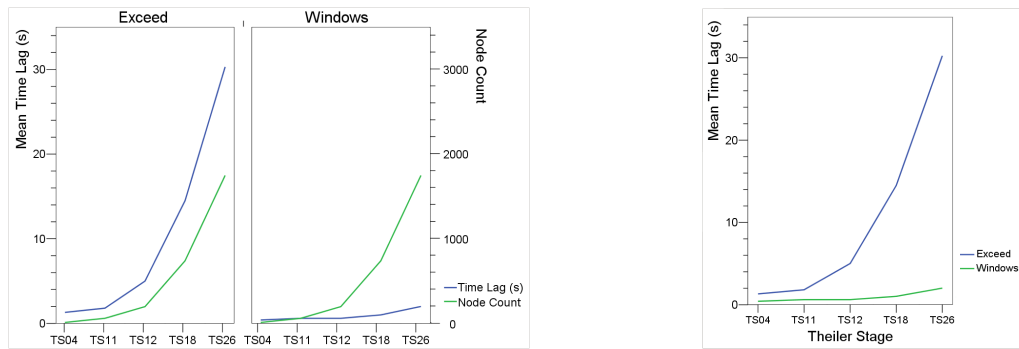


Figure 7.3. After improving the algorithm for laying out the graphs system response in Windows improved significantly; there was however little improvement in remote execution of the program.

An explanation for poor response and improvements for rendering of graphics in Java Swing applications during remote access are detailed in the 2001 reports by Sun Microsystems: *Java 2 Platform, Standard Edition v 1.4 Performance and Scalability Guide*¹ and *High Performance Graphics: Graphics Performance Improvements in the Java™ 2 SDK, version 1.4*². The applications started development in Java 1.4 then were continued in Java 1.5. However, to allow the browsers to be used on older systems features in Java 1.5 not backwards-compatible with 1.4 were not used.

These results pose a significant problem; humans use short-term and working memory to process perceptual input and solve problems [166]. Significant delays in system response lead to a decay in the information held in memory, requiring larger effort to perform tasks while users rehash plans to goal completion. This often leads to annoyance and an increase in the probability and frequency of errors occurring [16], especially for interactive systems such as the visualisation browsers under test. Even more significant than long response times is large variation in response; users develop an expectation of system response based on experience [166], so that variations in response may have a more detrimental effect on user productivity and satisfaction than long but uniform response times. The problems posed by the delays in system response with data load were reflected in answers to questions on the effects of variation in system response and from user reactions recorded during the evaluations. § 10.4.2 suggests options for improving drawing performance and interactivity.

Adaptability of humans means that workarounds for such problems will be sought [16], often employing shortcuts to goal completion and minimising interaction with systems. This has the associated disadvantage in a reduction in willingness to explore system functionality [165]. Providing feedback on system progress especially for long waits keeps users informed and reassured, decreasing frustration [166], and giving the opportunity to perform other task-related activities while waiting on system response. It would also be useful to provide options for undo and other error management as applicable, and/or the ability to terminate such processes where this would not result in system faults or data corruption, and without

¹See <http://java.sun.com/j2se/1.4/performance.guide.html> (last viewed Jul 2006)

²See http://java.sun.com/products/java-media/2D/perf_graphics.pdf (last viewed Jul 2006)

having to shut down the application.

7.3 Implementation of evaluation procedure

7.3.1 User backgrounds

Educational backgrounds and current field of work

The evaluation was carried out with ten users from the two main target user groups: researchers in biology and bioinformatics — the primary target, and in computer and other sciences working in bioinformatics (secondary targets). There was a degree of overlap in user backgrounds; for the purposes of this evaluation users were categorised into one of the two distinct groups, with the decision made weighted by educational background and current and former fields of work.

Biologists were expected to have a wide range of skill in computing, from basic to fairly advanced with some knowledge of programming. Computer scientists on the other hand were expected to have very little to general knowledge of biology, being concerned mostly with the development of tools for the analysis and presentation of biological or bioinformatics data.

Six users were researchers at the MRC, and four were MACS research students. Nine out of the ten users were at the time of the evaluation doing research in or working in a field related to bioinformatics; the last user was an HCI expert in MACS. Seven users were classified as biologists, five of whom have degrees in biology or bioinformatics and are currently working in these fields. One has an undergraduate degree in biology but a postgraduate in computer science and is working in bioinformatics, and the last has a degree in engineering and is working in bio-engineering. The other three users were classed as computer scientists: two have degrees in computer science and are doing research in this field, one with application to bioinformatics. The last user has an undergraduate degree in chemistry, a postgraduate in computer science and is currently doing a PhD in computer science. Figure 7.4 summarises users' backgrounds.

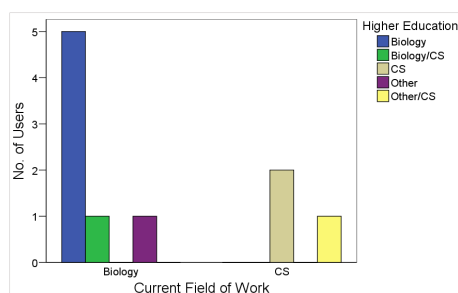


Figure 7.4. User backgrounds, recording education and current work

Age and gender

Five out of the ten users were female. Five were aged between 20 and 29, three between 30 and 39 and 2 were over 39 years old. Neither age nor gender appeared to have a significant impact on user ability, judging from observations made during the evaluations and user responses to the post-evaluation questionnaires. Users' research backgrounds were observed to have a more significant impact on interaction with the prototypes, especially when searching for specific information, and in user reactions to the implementation and labelling of functions.

Experience in computing

Figure 7.5 compares users' backgrounds with general familiarity with computers and computer systems.

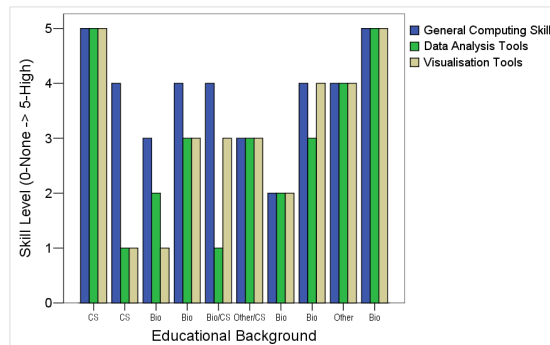


Figure 7.5. Computing skill and educational background of each user

Users were also asked to record their experience with input and output devices, storage media and basic microcomputer applications (refer § C.3.1 and figure D.1.1). All users made regular use of at least the commonest input devices, including the *mouse* and *keyboard*, and half the user group had additionally made use of *touch screens* and *track balls*. *File management systems*, *text editors*, *word processors*, *electronic spreadsheets* and *graphics systems* recorded regular use for most participants.

Range of specifications for users' computers

Each evaluation was carried out in the user's normal working environment (offices or research labs at the MRC or MACS); the visualisation browsers were therefore tested on a variety of systems. Operating systems (O/S) normally used range from Microsoft Windows® (NT, XP, 2000) to Apple Mac® (OS X, 9) to Unix (Linux, Irix®). Some users work on only one O/S, others work with Windows and/or Mac and Unix. Distribution of computing resources during the evaluation is shown below. All computers were resident on a network.

Users	O/S	CPU	Memory	Disk space	Monitor
4	RedHat Linux 8.0	P4 1.7GHz	256MB	40MB	17in, 1024*768
5	Windows 2000	P4 2.0GHz	1GB	80MB	19in, 2560*1024
1	Unix	PII 300MHz	unknown	unknown	19in, 1280*960

Internet use

Both online and standalone use are envisaged for the visualisation prototypes; it was therefore necessary to verify Internet connections available to users and use of web browsers (see figures 7.6 and 7.7).

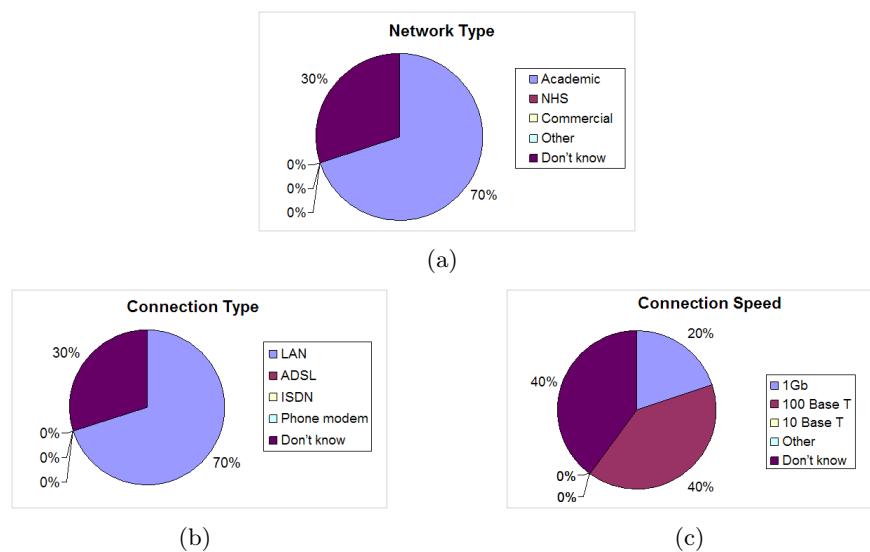


Figure 7.6. Chart 7.6(a) shows the number of participants using each of the network types available. Charts 7.6(b) and 7.6(c) show the distribution of users with access to the different Internet connections and speeds in common use.

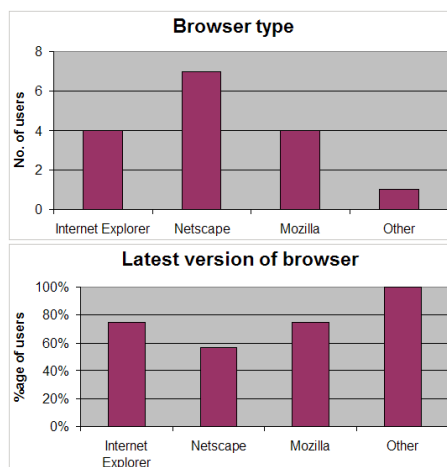
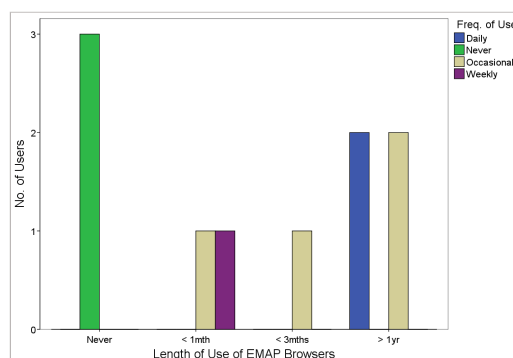


Figure 7.7. The chart at the top shows the different web browsers used (note that some users make use of more than one browser). The bottom chart shows the proportion of users, out of the total using each type of browser, using the most recent version of the corresponding browser at the time of the evaluation.

Experience using EMAP browsers

Figure 7.8 records length and frequency of use of the EMAP browsers. Three out of ten participants had never made use of the EMAP browsers prior to the evaluation. Use varied for all other users from occasional to regular, daily use.

Figure 7.8. The chart shows three users with no experience of the EMAP browsers while two use the browsers daily. All other users make occasional use of the browsers, for lengths of time from less than one month to over a year.



7.3.2 Evaluation procedure

The purpose of the evaluation and the methods for collecting user data were explained to each user, prior to presenting them with the instruction sheet and consent form (refer § C.1). The pre-evaluation questionnaire was then administered, to collect user background information. A brief demonstration of the functionality available for analysis using the visualisation browsers was then made, and users were given the opportunity to explore the prototypes before performing the tasks required. Users performed all tasks in the same order. It was noted that this had the potential of biasing (apparent) ease of use of the 3D browsers. However additional functionality in 3D that could not be implemented due to the space restrictions of 2D meant that carrying out tasks logically followed this order.

Because the help files had not been completed when the evaluation was carried out verbal responses were provided to user questions, without prompting users on specific actions to perform. Confirmation of actions required to complete a (sub-)task were, however, given where specifically requested. Understanding of results of users' actions were confirmed or corrected as required, if this was sought. For cases where users had significantly exceeded MTC and did not appear to be able to complete a task, for cases where usability problems had already been identified but had not been corrected prior to the evaluation, users were prompted with suggestions for completing the sub-task presenting a problem, and all such instances recorded.

The timer/logger built into the browsers was used to record functions called (from the menus or toolbars), attaching a time stamp to each. To confirm completion of a task users were required to click on the timer on the bottom, left-hand corner of each browser. Flow diagrams were used to record users' paths to complete each goal, supplemented by records of users' (physical) reactions, comments and errors made, requests for help and responses

to users' questions.

After completing the tasks required users filled in the post-evaluation questionnaire, followed by the SUS. A short, oral debriefing session was used to elicit any other comments users had, and to discuss especially those functions users had difficulty with. Participants were then thanked for their time and, for those who wished to receive further information on the development of the prototypes, confirmed that this would be made available in due course.

Three exceptions to the evaluation procedure are detailed below:

1. An oversight meant the user with ID 01 did not complete the SUS questionnaire. Having a small sample size, however, it was still useful to retain this user's results.
2. Availability of users 04 and 05 resulted in the two users performing the evaluation together. They, however, filled in the questionnaires independently. Interaction between the two users during the evaluation meant that they needed to refer to the evaluator less often for help, as they conferred with or prompted each other when unsure how to proceed or an error was made. Since this is normal practice in learning how to use a new application, and again, because of the small sample size, these results were not discarded. Additional analysis of the two users' results did not reveal significant differences from those of other users. Figure 7.9 compares task completion times for users 04 and 05 to the mean for all users. As for plots for SUS scores and overall satisfaction ratings (refer figures 7.10 and 7.11), differences in results are not found to be significant.

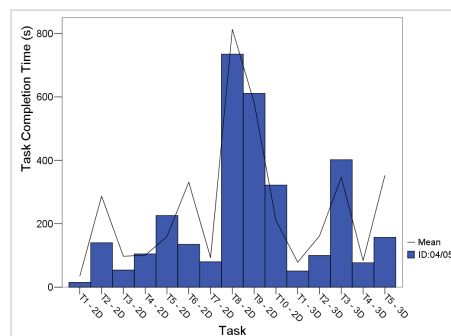


Figure 7.9. Task completion times for users with IDs 04 and 05 does not vary significantly from mean task completion time (for all users).

3. User 07 did not receive a demonstration of the prototype prior to performing the tasks. This oversight was recognised during the evaluation: the user was having difficulty determining functionality required to complete the tasks required, and task completion times were significantly longer than had been observed with previous users. The evaluation was suspended and the user given a brief demonstration of the functionality available. (Relative) task completion times did reduce significantly after this; it should however be noted that comments made by the user indicated that learning as the evaluation progressed also contributed to this.

7.4 Analysis of evaluation results

The small test user population means that statistical analyses may not provide reliable indications of usability of the applications developed. Testing with users from a wider group may have increased statistical reliability but with the much larger danger of reporting results not completely representative of the target user group. Statistical analysis of the data was therefore restricted to simple calculation of means within a 95% confidence interval (CI), and maxima and minima of result sets. Qualitative feedback, to support the quantitative data recorded, was obtained from user responses to the pre- and post-evaluation questionnaires and from observation of users recorded during each evaluation.

7.4.1 SUS Scores

The SUS scores obtained, shown in the plot in figure 7.10, show eight out of the nine users who filled in the questionnaire with a score above the middle mark (50). The highest score obtained was 90 (out of 100) and the lowest was 37.5. The 95% confidence interval for the mean, 62.5, has lower and upper boundaries of 51.5 and 73.5 respectively.

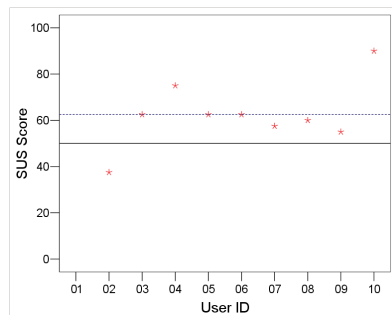


Figure 7.10. SUS score for each user. Note the user with ID 01 has no SUS score. The average for the SUS is therefore recorded over nine users, shown by the broken line.

7.4.2 General satisfaction ratings

Figure 7.11 compares overall user satisfaction for the 2D and 3D browsers, calculated by finding the mean rating on the Likert scale from 1–9 for questions in sections 3 to 7 in the post-evaluation questionnaire (refer § C.3.2). (Note that questions with a response N/A were disregarded in calculating each mean.)

The overall satisfaction ratings for individual users show eight out of ten users with mean rankings for both browsers above the central mark 5, with five of those values lying above the mean for the entire sample. One user ranked the 3D browser above the central mark and the 2D below, and the last ranked both below the central mark. Half of the users ranked the 3D browser higher than the 2D. Mean rankings for the 3D browser were higher than for the 2D, both over the whole user population and by user group, as figure 7.12 shows. Results also show biologists, the main target, on average rated higher usability for both browsers; the two lowest individual means were CS researchers. There is also a relatively large variation in results of CS users, which may be due to the very small number of users falling in this group.

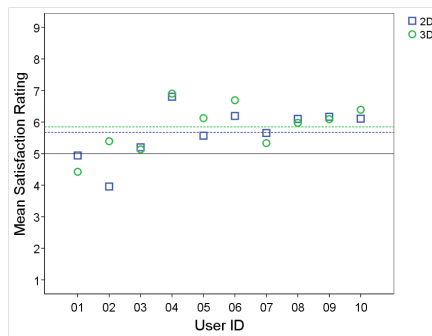


Figure 7.11. Overall user satisfaction rankings for each of the visualisation browsers; overall the 3D browser was found to be more usable than the 2D.

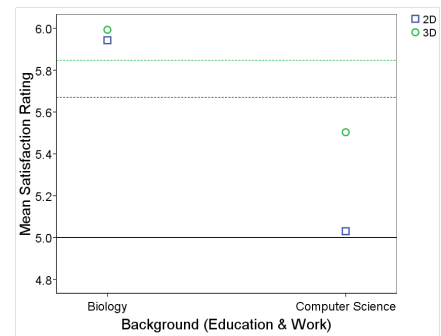


Figure 7.12. The broken lines show both groups on average found the 3D browser more usable than the 2D, with biologists generally recording higher usability than CS researchers.

Charts detailing mean ranking for each item in the post-evaluation questionnaire can be found in § D.2.1 and D.2.2, and values for overall mean rankings are as below:

Overall means

2D: 5.67 [95% CI: 5.09–6.25]

3D: 5.85 [95% CI: 5.30–6.40]

Biology only

2D: 5.94 [95% CI: 5.42–6.55]

3D: 5.99 [95% CI: 5.35–6.42]

CS only

2D: 5.03 [95% CI: 2.25–7.80]

3D: 5.50 [95% CI: 2.68–8.32]

The following items recorded rankings in the top ten for both browsers (not necessarily in this order):

- quietness of system (as a measure of how busy system was)
- consistency of terms used
- good relation of terms to users' normal work
- consistency in system messages
- eased ability to determine lineage using the visualisations
- advantages provided by the visualisations over the text (for analysis)
- time taken to perform tasks
- improvement in data analysis over EMAP browsers (for tasks performed).

For the 2D browser alone the following were also ranked in the top ten:

- hiding of sub-trees to reduce occlusion
- reliability of the system.

The following also fell in the top ten items for the 3D browser:

- advantage provided over the use of text boxes for tracing lineage

- options for zoom that aid reduction of occlusion.

Items with the ten lowest rankings for both browsers include:

- large variation in system speed (recording the lowest ranking for all items for both browsers)
- occlusion of data
- average time to perform tasks
- (in)flexibility of system
- ease of navigation through data
- level of system support for error recovery
- frustration using system
- rigidity of system.

Additionally, the 2D browser recorded poor rankings for the following items:

- ease reading text on screen
- ghosting of nodes
- occlusion of data especially in the layout with a vertical orientation.

Other items with low rankings for the 3D browser were:

- ability to identify errors and sources of errors
- difficulty of system
- good balance between catering for needs of experienced and inexperienced users.

7.4.3 Assessment of the 2D browser

Overall user reactions

A breakdown of the satisfaction ratings into the five sub-groupings examined by the post-evaluation questionnaire (refer § 7.1.2) shows a mean of 4.78 for *Overall reactions to the system*. A more detailed analysis of the different aspects of usability studied, in the following four sections, identifies functionality that provided useful contributions to data analysis, some of which recorded improvements over methods already available for analysis. Functionality that was not found to be very useful was also pointed out. Analysis of user comments provided additional information that was used for further design and development of improved functionality for analysis.

Means for each category for which usability was measured are as below:

Overall reactions to the system:	4.78 [95% CI: 4.23–5.32]
Data Visualisation & Screen:	5.86 [95% CI: 5.38–6.34]
Terminology & System Information:	5.93 [95% CI: 5.11–6.75]
Learning:	5.67 [95% CI: 5.05–6.28]
System Capabilities:	5.92 [95% CI: 4.78–7.05]

Data visualisation and screen

The mean ranking for all items in this group was 5.86, with more than half the items ranked above the mean. (High) occlusion recorded the lowest ranking, with a mean of 3.00. Users also recorded difficulty reading text on the screen and navigating through the data. Comparing the three layouts, the vertical or top-down (TD) had the highest mean ranking for ease of use at 6.60, while the horizontal or left-right (LR) had a mean of 6.50. The radial layout scored lowest, at 5.00. Occlusion was ranked below the central mark: 4.7 for LR, 4.5 for the radial and 4.4 for the TD, confirming occlusion to be a significant problem. Users ranked ease of locating information required at 5.11.

Compared to the text indices usability of the visualisations was ranked as high (7.38), with data structure (6.71), understanding of data (6.86), search and query (6.57) and tracing user paths and lineage (7.43) all above the overall mean for this sub-grouping. Creation of groups was ranked at 6.14.

Hiding of labels and sub-trees recorded relatively high rankings (6.78 and 7.10 respectively) for their contribution to the reduction of occlusion. Zoom had a mean ranking of 6.5, and switching between layouts, 5.6. Ghosting was found not to be very useful for reducing occlusion, with a ranking of only 4.38; functionality for ghosting was not well developed at this stage so that this was to be expected (see figure 6.15(c)).

Terminology & system information

Relation of terminology to users' normal work recorded the highest mean ranking at 7.57, use of terms and consistency of system messages followed with scores of 7.20 each. The lowest rankings for this sub-grouping were for the level of system support for error recovery (4.00), and for users' ability to identify errors and their sources (4.56).

Learning

(Relative) time required to carry out tasks recorded the lowest ranking at 3.80. Amount of time users felt they required to learn how to use the system however recorded a mean ranking of 7.1, indicating the system would take on average a very short time to learn to use. The longest time estimated to learn to use the system was one month, the shortest one day, and the average fell between one and two weeks.

Relation of terminology to users' normal work was ranked at 6.50, communication from the computer just above the mid-point at 5.11, and all other functions recorded rankings around the mean (5.67) for this sub-grouping.

System capabilities

This sub-grouping recorded overall a mean ranking of 6.57. (Low) system noise was ranked at 8.29. Compared to the EMAP section browsers this system recorded a ranking of 7.00 for ability to provide simplified data analysis. Reliability of the system was seen to be

fairly high, with a mean ranking of 7.00. Variations in system speed however recorded an average ranking of 2.86, the lowest for the evaluation. Users also judged system speed on average to be low, at 4.30. Balancing well the needs of both experienced and inexperienced users was ranked at 4.78. Ability to correct mistakes recorded an average 5.56, and level of functionality provided by the system was ranked at 6.70.

7.4.4 Assessment of the 3D browser

Trends for rankings for the 3D browser follow those for the 2D, although actual rankings were on average higher than for the same items in 2D. Looking at *Overall user reactions*, which recorded an overall mean of 5.10, here, as for the 2D, users ranked the system being frustrating and being rigid lowest. The 3D browser also received poor ratings for being difficult, but was ranked higher than the 2D for providing adequate power to users. Finding the system stimulating again received the highest mean ranking of 5.22 for the 3D browser.

Rankings for *Data visualisation and screen* were in general much higher than for the 2D browser, with an overall mean of 6.19. Only two items fell below the central mark - navigation through the data and occlusion, at 4.20 and 4.30 respectively. More than half the items in this sub-group lay above the mean of 6.19. Improved ability to trace lineage was ranked highest at 7.29, followed by advantages of the system for analysis over the textual indices, at 7.25.

Trends for *Terminology & system information*, *Learning* and *System capabilities* were also very similar to those for the 2D browser, but again with higher actual values. Means for each category for which usability was measured are as below:

Overall reactions to the system:	5.10 [95% CI: 4.40–5.80]
Data Visualisation & Screen:	6.19 [95% CI: 5.66–6.71]
Terminology & System Information:	5.97 [95% CI: 5.19–6.75]
Learning:	5.73 [95% CI: 5.13–6.32]
System Capabilities:	6.20 [95% CI: 5.22–7.17]

7.4.5 Task completion times

Figures 7.13 and 7.14 show mean time to completion for each task in each of the 2D and 3D browsers respectively, compared to the MTC for each task. Learning was exhibited by users performing repeated tasks in 3D in shorter times than expected. A significant example is for grouping of nodes — tasks T8-2D (task 8, 2D browser) and T3-3D (refer figure 7.16); users also found grouping in 3D to be more intuitive than for the 2D browser.

Figure 7.15 compares completion times for each user for each task to MTC, which shows times for most users for tasks T2-2D, T3-2D, T7-2D, T1-3D and T2-3D to be very high compared to MTC. Analysis of the flow diagrams for each evaluation and comments made during the course of the evaluation provide clues that may explain these results. Most of the problems identified were found to be due to difficulty identifying functionality required to carry out tasks, mainly due to function labels whose meaning was not immediately obvious.

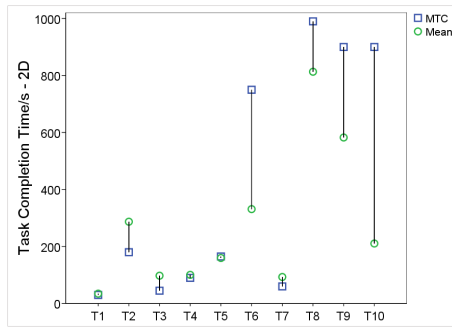


Figure 7.13. Mean completion time for each task for the 2D browser, compared to MTC

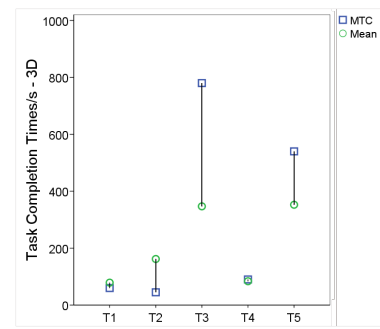


Figure 7.14. Mean completion time for each task for the 3D browser, compared to MTC

Additional textual information provided to describe objects in the scene did not always contain enough information to allow users to draw conclusions about data with confidence. Another problem was restrictions to data input that made editing of the graph structures tedious.

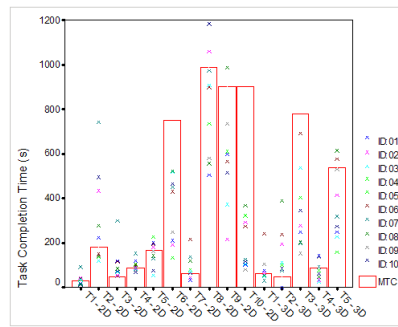


Figure 7.15. Individual task completion times for both browsers, compared to MTC

User task completion times exceeding MTC

T2-2D: determining lineage within a stage of development T2-2D required users to identify a specific component and bring up textual detail for the components derived directly from it. Three users took significantly longer than MTC to complete this task, with the worst case being almost 4 times longer than expected. However this occurred for the case noted previously where the user had not had sufficient preparation for the evaluation (refer § 7.3.2), which might explain the anomaly.

The next longest completion time occurred because the user's understanding of the function for determining lineage did not map to the system implementation, highlighting a problem with ambiguity in terms used to describe lineage, and also noted in other instances and for other users. The next result, also significantly large, was due to implementation for the display of sub-trees when expanded in the default window. Users mostly expected the number of levels drawn to be increased automatically on expansion of a sub-tree; however the original implementation hides all nodes beyond the current limit for levels drawn to the screen, even for sub-trees expanded beyond this point.

T3-2D: retrieving additional information on data objects Users were required to determine if a specified set of previously identified nodes had synonyms. The simplest method for completing this task is to display textual detail for each component. Completion times for all users for this task were greater than MTC. This highlighted a problem with presentation of textual information on components: the (component detail) dialog boxes originally listed only component properties with non-empty values. Users therefore had difficulty determining whether values existed for component properties but were not available, or if these values were null, or the attributes in question were defined for the data nodes. Users varied between searching further for the information required and/or asking whether values for attributes existed or could be retrieved using alternative functionality.

T7-2D: search for and highlight component(s) of interest The optimal solution to this task is to perform a search on the component name. One user took almost four times the MTC. Expecting search results to be non-volatile, the search dialog was closed before recording the results, repainting the graph and clearing out search hits, so that the task had to be repeated.

Two other users recorded more than double the MTC. Both users first made an unsuccessful attempt to identify the components by visually scanning the graph before attempting a text search to highlight the components required.

T1-3D: loading multiple ontologies into 3D browser T1-3D required four ontologies to be loaded into the 3D window. One user took four times the MTC to complete this task and a second user took almost double the MTC. Each of these users loaded the data sets individually even though the open file dialog in 3D allows multiple selection; this highlights a problem with making users aware of the functionality available (the 2D browser only allows single selection because graphs are loaded in separate frames). It should be noted that the longest time recorded was also due to the user examining each graph after loading; this user spent a significant amount of time exploring the visualisations generated and functionality available, taking longer on average than most users to perform most tasks, but with the second highest mean ranking for usability.

T2-3D: loading and unloading ontologies One user overlooked this task, but all others took significantly longer to perform the task than was expected mainly because of differences in implementation in the two browsers. Users expected context (popup) menus to be provided as for the 2D browser. However because all three mouse buttons are used for navigation in 3D, popup menus are not available.

A second problem was difficulty selecting the graph to be unloaded. It was not immediately obvious that selecting the root node was necessary to select a tree. Further, some users recorded difficulty selecting the nodes representing roots of some graphs.

A third factor was that the last tree loaded could not be seen from the default viewpoint;

users were required to zoom, translate or rotate the visualisation to bring this graph into the viewing area. Unfamiliarity with the navigation controls contributed to an increase in task completion time. Also, most users expected the stages to be (re)ordered by stage number and were therefore not sure whether the load had been successful or not, especially because the new graph loaded was not immediately visible.

The largest task completion time was 8.6 times that of the MTC. This anomaly was due mostly to an unexplained increase in system response time and poor O/S response that might have been due to a network slowdown.

User task completion times significantly lower than MTC

T6-2D: locating a specified component and determining lineage T6-2D was expected to have a large completion time; it was necessary to identify a specified component and its sub-parts, and retrieve additional detail on each. The task was complicated by high occlusion in the ROI at default zoom. Users however performed this task significantly faster than expected by making use of functionality for zooming into ROIs.

T8-2D and T3-3D: creating groups T8-2D and T3-3D, involving the creation of new *groups* were expected to have large completion times, especially because the functionality available for editing the graph structures was not very well developed. Learnability of the system was demonstrated by an average decrease of 56.54% in task completion times when this exercise was repeated for the 3D browser, compared to the expected 21% decrease (see figure 7.16). Users also found it easier to identify nodes of interest in the 3D browser using the visual structures, allowing more intuitive creation of groups.

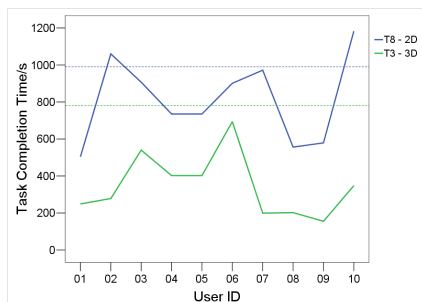


Figure 7.16. The broken lines across the graph represent MTC for each of tasks T8-2D and T3-3D, creating *groups* in 2D and 3D respectively. All users took less than the MTC when the task was repeated in 3D, with most users taking significantly shorter than MTC. The difference between MTC and actual completion times was smaller for the 2D browser, which had two users exceeding MTC.

T9-2D and T10-2D: working with a large data set in 2D Tasks T9-2D and T10-2D required analysis using TS26 which contains 1748 components (almost 30 and 10 times more than TS11 and TS12 respectively). Drawing all nodes, links and labels to the screen resulted in poor system response (see also figures 7.2 and 7.3). However all users completed T10-2D in less than half the MTC. Learning may have contributed to this: both tasks required searching for a specific component, then carrying out a task based on the properties of the component of interest. The only significant contribution to delay completing these two tasks, from user observation, was system response time.

7.5 Discussion of evaluation findings

The results provided information that could be used to answer the questions posed at the start of the evaluation, but also brought up several others that needed to be resolved in order to determine what would be usable methods for the analysis required.

Users found overall that the visual overviews of the ontology data aided determination of data structure, improving searching for data of interest and IR. Quantitative results were supported by user comments during the evaluation and recorded on the post-evaluation questionnaire, indicating advantages of the visual system over text-based data analysis. Despite limited functionality for analysis in 3D and greater difficulty recorded in navigation, the 3D browser was rated higher on average for usability than the 2D; the 3D visualisations were described as more intuitive. The results are tempered, however, by the poor system response exhibited in the 2D browser during analysis of the larger data sets. Users also found the system fairly easy to learn, quickly identifying functionality required to carry out tasks.

Restrictions in space in 2D mean that the 2D browser cannot satisfy requirements for simultaneous analysis of multiple data sets (refer § 5.3); occlusion limits the amount of data that can be displayed effectively. Only one data set is drawn in each 2D frame, allowing (detailed) analysis of individual data sets in isolation. The main advantage in the extension to 3D is the additional dimension that provides more space in which to contain and display data, regardless of screen size, and bounded only by resources required to build objects in the scene. Using 3D makes it possible to perform simultaneous, visual analysis of multiple data sets, to retrieve relationships across different ontologies. Data encoding in the 3D browser was also found to be more effective, making it easier to interpret data attributes based on colour of nodes and links than in the 2D browser.

Significant increase in system response time with data load in the 2D browser constrained the amount of data that could be drawn to the screen before interactivity became severely limited. Data load had little effect on system response for the 3D browser; this factor may have contributed to the higher usability rankings for 3D recorded during the evaluation.

Domain knowledge had significant influence on methods used for searching: biologists, with prior information on data content, quickly formed effective mental models of data structure that allowed intuitive use of the visualisations for locating data of interest. CS users on the other hand were more dependent on the search dialog to locate nodes of interest, performing what was in essence *blind* searching. These results are similar to those in [44], who found that ability in interactive visualisation systems is dependent on interpretation of data — the semantic models users form of data structure. [181] additionally observed that users with high spatial awareness are able to navigate more effectively through visual structures, resulting in more intuitive analysis and IR. 3D may be used to provide an information space that harnesses perceptual ability in humans to improve analysis; careful design that

takes into account user backgrounds, knowledge and ability, mapped to design principles in *Human Factors* and *HCI* should enable useful abstraction of especially large, complex data sets [151], making it easier to become immersed in data, encouraging exploration and leading to a better understanding of data structure. Research also supports evaluation findings that the provision of additional cues, both graphical and textual, serve as an aid [39], especially for users with little domain knowledge and who are more likely to encounter disorientation while navigating through data, especially where they exhibit low spatial awareness [44].

The evaluation results provided justification for further development, especially for the 3D browser; the findings support research into spatial awareness and perception that record increased ability of humans when performing visual data analysis. Development to improve navigability in especially the 3D window is important to encourage data exploration, and to provide greater support for locating and identifying data of interest and relationships within data. [39] and [161] stress the importance of support for navigation through data, in addition to development that matches user ability and work environments, to obtain effective analysis.

Chapter 8 details improvements to the visualisation browsers as a result of the usability evaluation described in this chapter, then presents design for new functionality, based on further research into the effects of human perception and spatial awareness on the ability to perform detailed analysis of especially large and/or complex data sets.

Chapter 8

Visual solutions for data analysis

The usability evaluation described in chapter 7 provided confirmation that the visualisation browsers implement functionality that aids analysis of the anatomy ontology data under study. There are, however, recognised limitations of the graphs used to visualise the data; this chapter details changes and additions to functionality suggested by an analysis of the evaluation findings. This leads to further development of the techniques described in chapter 6 to satisfy more completely the requirements for data analysis and IR.

8.1 Changes to prototypes based on evaluation results

8.1.1 Graph attributes and layout

Data encoding

The DAGs currently use a single shape of fixed size to represent each data type. Even though it is acknowledged that variation in shape and size of data objects would provide more options for encoding data attributes only a limited set of options are made available. An important design consideration was to use simple visual representations of the data to minimise complexity and clutter in the graphs generated.

Contrary to all other users who found colour coding more effective in the 3D browser one user who was colour-blind recorded difficulty distinguishing between some data elements because of the colour combinations used. Added complexity due to depth aggravated this problem during use of the 3D browser. Options for interactive modification or encoding of data attributes using size and/or shape are being considered to help resolve this problem. Current options provide an editable legend (in 3D) detailing colour used to encode data properties (refer figure 6.26).

Expansion of nodes

The number of levels displayed in the 2D graphs can be interactively modified using a range slider, from the minimum (of up to three) displayed when the data set is first loaded, to

the maximum for each graph. Expanding any node in the graph continued to hide nodes in its sub-tree beyond the current number of levels set to display. Users, however, expected expansion of a sub-tree to increase the number of levels in the graph automatically, if this was required to display the entire sub-tree of interest. Graph layout was modified to match user expectation: expanding a node now alerts users and increases the number of levels in the graph if necessary.

Ghosting

The original implementation (see figure 6.15) only faded out nodes when set to ghost out. This function now also hides labels and links into or out of ghosted nodes, resulting in more effective reduction in occlusion.

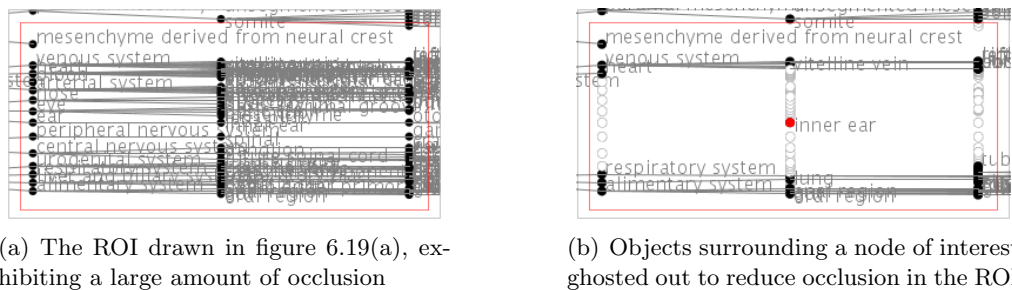


Figure 8.1. Improved implementation for ghosting out of nodes; this now hides labels and links into and out of nodes, in addition to drawing only the outline of ghosted nodes. The node of interest, the *inner ear* is easily identified in figure 8.1(b).

Mapping between windows

The ZoomPane Mapping between the main window and the (modal) sub-window that magnifies data sub-sets was improved by providing access to functions for editing nodes extracted to the sub-window. Figures 6.17 and 8.3 show the two context menus used to edit objects in the **ZoomPane**.

2D and 3D browsers Though base functionality implemented in the 2D and 3D browsers is the same they are run as separate applications. Users suggested linking the two browsers, to allow isolated analysis of a sub-set (from 3D to 2D) or to widen the scope of data analysis (from 2D to 3D). The browsers now provide the option to switch directly between the two views, maintaining the system state and graph structures already drawn to the screen, so that continuous analysis can be performed from the alternative perspectives.

8.1.2 Supplementary textual detail

Component detail

The component detail dialog originally displayed only data attributes with non-empty values; users however had difficulty distinguishing between attributes not defined for an object

and those with empty values. The dialog now lists all properties defined for each object type, returning *None* or *Not Available* for attributes with null or empty values.

To enable annotation of data, user comments may now be attached to specified components or user-created mappings between components. Comments may be saved with all other ontology data to a *system state* (XML) file (a sample of which can be found in § A.3).

Data labels

Labelling of nodes in the graph alternates between displaying labels on the canvas and *popping up* the label for each node as it receives the focus. Suggestions were made to *pop up* labels even when node labels are displayed in the graph; some users felt this would help to distinguish labels in areas with high occlusion, where they may be rendered illegible, or for cases where labels run off the edge of the window.

Some users commented that hiding of all labels was not very useful as this made it difficult to tell what the data represented. There are now two options for hiding labels — ALL labels may still be hidden, or only selected labels are hidden. The latter is useful for reducing occlusion in an ROI, while still providing enough information to easily identify data nodes. Hidden labels are still *popped up* for the component with the focus.

Glossary

A glossary of terms was suggested to aid both computer scientists and biologists in verifying the meanings of terms or functions. The help files which can be accessed from either browser now provide brief descriptions of all functions implemented.

8.1.3 Editing data structure

Creation of Groups

Most users had difficulty locating the *Create Groups* function, which was initially placed in the *View* menu. A *Grouping* sub-menu is now available from the *Edit* menu to map to users' semantic interpretation of this function; users expected to *edit/create* a group, not simply *view* one. Functionality for editing and removing previously created *groups* has also been implemented, accessible from this sub-menu.

Creation of a *group* node now requires users to set a primary parent; so that a *print name* (fully qualified name) can be determined, necessary to map a (default) path to the root of a graph. This allows automated retrieval and tracing of lineage through a single ontology or across multiple data sets.

Existing nodes in the graph may be selected to form part of a *group* by entering node IDs directly in the *grouping* dialog shown in figure 8.2, the preferred method for users with a good knowledge of data content. The scrollable list available for selecting components to form a group has been replaced by a drop-down list. Users had difficulty switching between

multiple selection in the graph and the list; the latter required depressing the *Control-key* to retain previous selections (following convention for multiple selection in lists for standard GUIs), while this was not necessary in the graph. Although the drop-down list allows only single item selection it does provide independent selection of nodes in the graph.

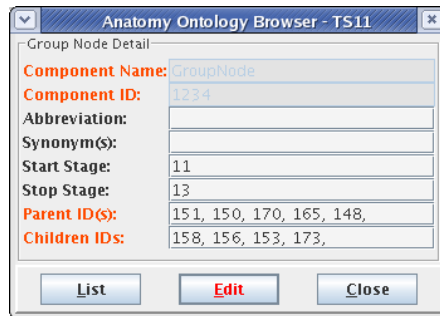


Figure 8.2. A custom dialog being used to edit an existing *group node*. In *edit mode* the component name and ID can no longer be changed (and have been greyed out). Fields whose labels are highlighted must be filled.

Users also requested more direct feedback (in the form of a list) on components selected to form part of a *group*; highlighting in the DAG is not always easy to discern. Currently this is provided by the list of node IDs in the *grouping* dialog.

In order to be able to work with only data of interest the need to suppress all nodes outside *groups*, or view *groups* on their own was recognised. User-created *groups* may now be redrawn in the **ZoomPane** shown in figure 8.3, allowing analysis in isolation.

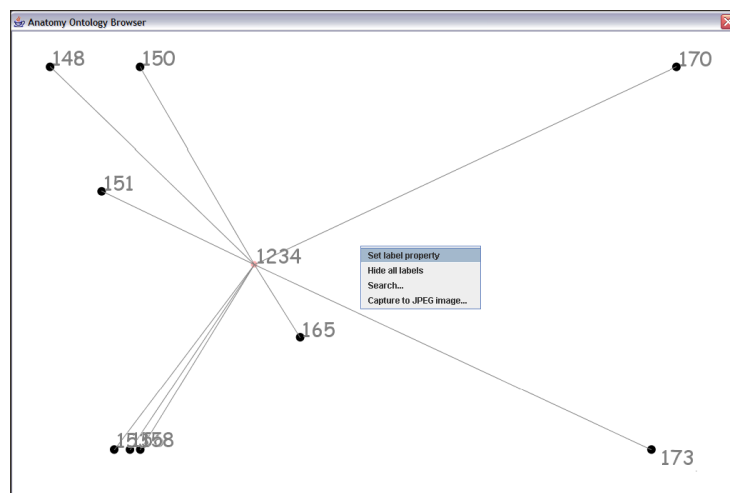


Figure 8.3. Viewing the nodes that make up the user-created *group* shown in figure 6.29 in isolation in a 2D window. One of the pop-up menus available for editing the data is displayed.

8.1.4 Search options

Widening search

To provide a larger number of options for IR the search dialog now provides the option to widen search results on *component name* or *print name* to include matches for *synonyms* and/or *abbreviations*. Additional matches are highlighted in the search dialog and encoded in the graph using deep cyan.

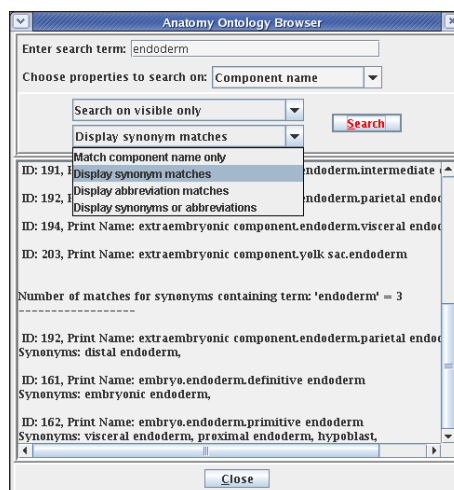


Figure 8.4. Additions to options for retrieving results in search dialog

Retrieving user comments

The option to search within user comments attached to nodes or to retrieve all nodes that have been annotated has been implemented.

8.1.5 Navigation and exploration

Undo/history functions

To encourage exploration especially in the 3D browser where disorientation easily occurs, it would be useful to be able to undo undesirable effects of actions. There is no history function currently implemented for either browser, but the 2D browser provides the option to reset a selection of nodes or a complete graph, removing all formatting applied to nodes and links. The 3D browser makes use of in-built functionality in Java3D to allow the viewpoint to be reset to the centre of the universe if users become lost in the 3D space.

The ability to save user sessions has been implemented, writing out a description of each graph to a reloadable XML file (see § A.3). This allows incremental analysis to use previously created *markers* employing annotation of nodes and user-created links, to aid users in data exploration. A snapshot of the current view may also be saved to a JPEG image.

8.2 Additional suggestions for changes to browsers

The following sub-sections discuss important suggestions for changes which have not yet been fully implemented. These include those suggestions which cannot be implemented effectively because of limitations in technology available or that would not provide advantages over solutions already available.

Overview map

Because the overview is lost in the 2D browser when the graph is zoomed beyond the default (100%), an overview map would help to maintain a sense of location within the data structure. This would be even more useful in 3D where disorientation easily occurs while navigating through the data structure.

The **ZoomPane** could be used to hold such an overview. However being smaller than the main window, displaying all nodes in any but the smallest data sets would result in severe occlusion; folding away nodes to obtain data abstraction would defeat the purpose of the overview. Alternative methods for abstraction are required to provide a solution to this problem.

Re-ordering of nodes and graphs

Graphs were originally ordered by their load time in the 3D window. For tempo-spatial data such as the EMAP ontologies this may make it difficult for users to locate data of interest: user expectation was that graphs would be (re)ordered if required, to provide a timeline through the data. Alternatively, the ability to reposition graphs interactively would give users more control over visual structures in the 3D window, increase confidence in analysis results and result in visualisations that more closely match users' mental models of data structure. Interactive repositioning of nodes (in both 2D and 3D) was also suggested as an option that might be useful in reducing occlusion.

These suggestions however present two new problems: links drawn between graphs in 3D would have to be broken before re-ordering the graphs. A second more significant problem is that independent reordering of structures in the 3D window is compounded by challenges in navigation; it is difficult to regain an overall structure that still allows simple identification and mapping of relationships between data sets.

It should be noted that for multiple loads of data sets, trees are ordered by *treeID* before being drawn to the window, allowing a timeline to be obtained for data belonging to a single organism. Further, nodes in each sub-tree and level in a tree are now ordered by *component ID*, to ensure consistent layout of graphs (for related data sets).

Navigation controls

A final request was for the provision of on-screen navigation controls for the 3D browser, as a potential option for more intuitive navigation than (the default) using only the mouse and keyboard. This would however require a significant amount of design and development and is only being considered as an extension to the current browsers.

8.3 Solutions for open analysis issues

The 2D browser builds on proven methods for visual analysis, using node-link graphs to provide a spatial representation of the hierarchically structured biological ontology data. The graphs provide overviews of the text indices that aid analysis by helping users construct useful mental models of data structure; increased ability to perform analysis was confirmed by the usability evaluation performed (refer § 7.4 and § 7.5). To fully satisfy requirements for data analysis and IR additional functionality was required for intuitive identification of data of interest and relationships between data sets. This project developed a solution that layers the individual ontology graphs in independent 2D planes in 3D space (as described in § 6.7), reserving the third dimension for holding relationships that cross data sets (refer figures 6.27 and 6.30). The visual structures that result allow both analysis of individual trees in isolation and comparison between multiple data sets.

Simply being able to draw relationships using the extra space provided by 3D is not in itself enough to satisfy data analysis requirements; it is necessary also to be able to retrieve different types of equivalence between data element pairs across multiple data sets. Spatial and perceptual cues are required that improve the ability to identify locations in which relationships of interest may be found, and also what types of relationships are stored in the data. The rest of this section describes improvements to the browsers, based on a review of users' information requirements. Further research into the influence of human perceptual and cognitive ability on data analysis also fed into the development of the solutions to the limitations of current visual analysis.

8.3.1 Querying with a direct-manipulation interface

Marked differences in search strategies were identified for users, based mainly on background and domain knowledge (refer § 7.5); identifying cues for querying the data sets most effective for each of the two main target user groups is necessary to maximise the potential of spatial analysis. Commonly used visual cues include (semantically meaningful) *landmarks*; [150, 181] note usefulness of landmarks for navigation through data. Interactive generation of such markers would allow the browsers to incorporate domain knowledge of different users in building structures for continuous, incremental analysis, aiding the identification and retrieval of data of interest. Placing markers in data structures helps to reduce disorientation, providing *maps* that increase ability to navigate effectively through data [44, 181]. Supporting textual detail is also important in confirming results of visual analysis; spatial representation of data is not as effective in isolation [51].

Different types of equivalence have been identified for the EMAP and XSPAN data (refer § 5.2 and figure 5.4), describing among others, mappings based on similarity between *cell* and *tissue types*, *homology* (common lineage for components in different organisms), and *analogy* (components in different organisms with similar function). A graphical interface has been created that simplifies formulation of simple *AND* and *OR* queries to retrieve defined

relationship types. The dialog allows creation of new mappings between any node pair drawn in the 3D browser, and loading of existing mappings from a text or XML file (file structure is described in § A.2). Directional mappings are stored for each node pair, creating a lookup table that searches for mappings based on component IDs. (Note that mappings between component pairs belonging to different ontologies, though contained in **Relationship** objects are stored and handled separately from the *part-of* **Relationships** between components in the **AnatomyOntology** used to draw each graph).

Figure 8.5 shows mappings drawn in the space between three ontologies, using the custom dialog described. The query retrieves mappings stored that match the three types selected in the dialog. Search results may be restricted by selecting (text descriptions of) mappings (defined for each graph loaded) in the second dialog and moving them to the main query dialog (otherwise all mappings stored that meet search criteria will be drawn). § 8.3.2 describes options provided for visual presentation of search results.

8.3.2 Mapping equivalence across multiple ontologies

The simple query interface described provides improved retrieval and display of relationships between component pairs. The default, illustrated in figure 8.5, draws colour-coded links between node pairs visible in the browser that satisfy search criteria.

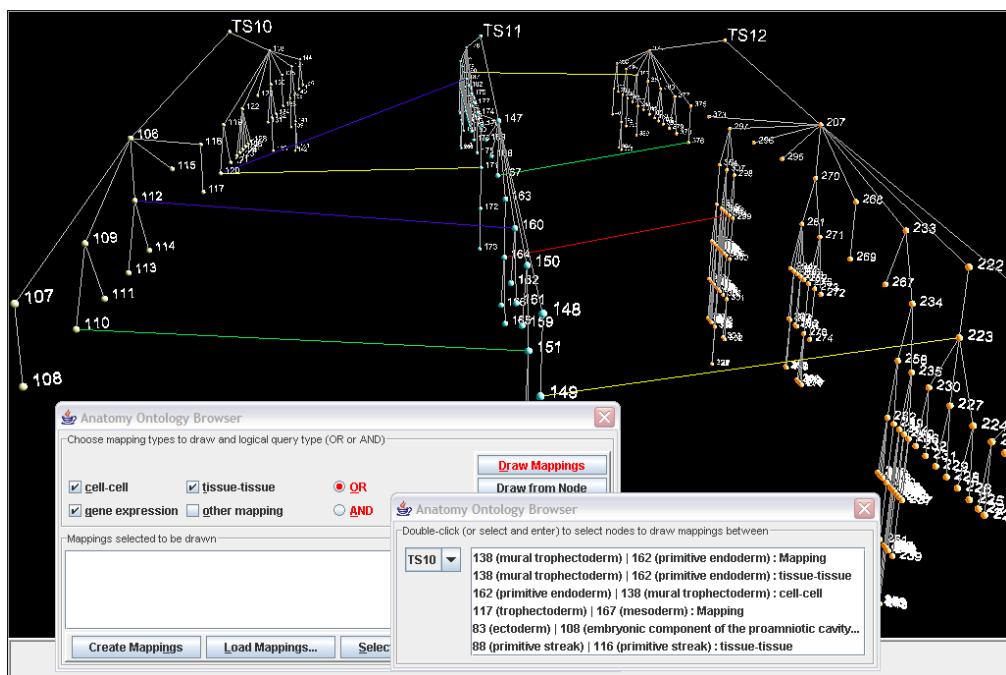


Figure 8.5. The dialog shown at the bottom of the window is used to build *AND* and *OR* queries that retrieve defined relationships across different data sets. Colour-coded links are then drawn between node pairs that satisfy search criteria. A sub-window of the mappings dialog is shown, listing mappings defined for nodes in each graph.

One link is highlighted (in red) by clicking in the graph. Additional textual detail may be displayed by double-clicking on any link or choosing the appropriate item from the *View* menu.

To satisfy requests for alternatives to drawing physical links between nodes users may now choose to draw a (colour-coded) wireframe round each node for which a valid mapping is found, as illustrated in figure 8.6. Limiting the number of new objects drawn to the screen removes distractions and allows greater focus on ROIs, especially useful for cases where a large number of mappings are stored. This also has the added advantage of preventing potential crossing of links drawn between trees. Functionality is also provided that allows uni- or bi-directional links into or out of a single node of interest to be drawn. To retrieve textual detail on each mapping identified the link of interest or either of its end nodes may be selected and the appropriate option chosen from the *View* menu.

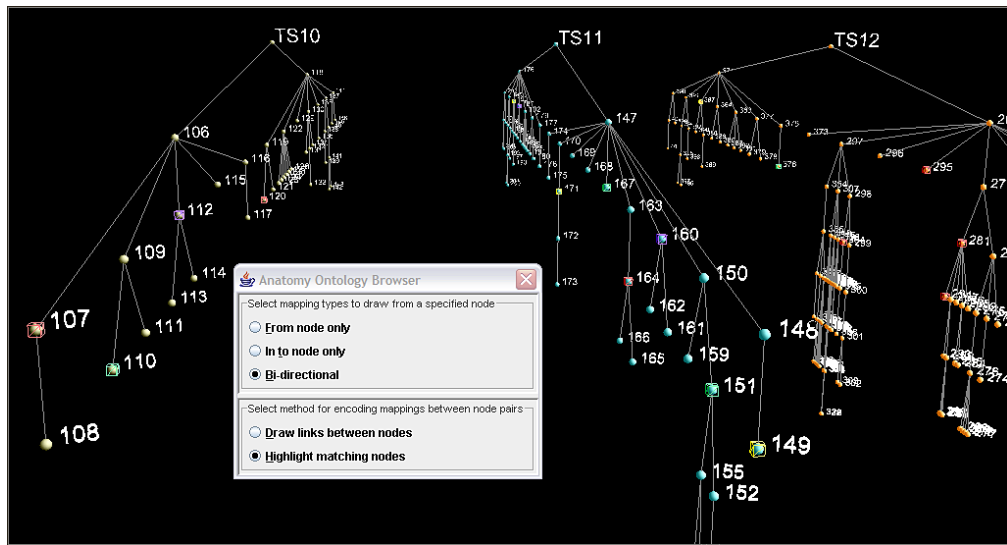


Figure 8.6. An alternative to links between nodes highlights each component for which a relationship is defined that satisfies search criteria using colour-coded wireframes. The snapshot shows the results of a query identical to that for figure 8.5, but suppresses links between nodes, replacing them with the wireframes shown.

8.3.3 Tracing lineage within and across ontologies

§ 6.7.5 demonstrates the first attempt at a spatial representation of lineage during development of a single organism; this method required users to identify components in successive ontologies through which a lineage trace was to be drawn. Further development was necessary to retrieve lineage automatically: this is now achieved by transparent searching through all ontologies loaded in the 3D window, to retrieve *print* or fully qualified names matching that of a component of interest for temporal data such as the developmental stages in an organism. The system then builds a **LineagePath** by sorting matches based on time of occurrence of each component. A polyline is drawn through the component list, tracing lineage for the component of interest through the ontologies displayed, as figure 8.7 illustrates. The list of components through which a **LineagePath** is traced can be brought up by double-clicking on any section of the path drawn.

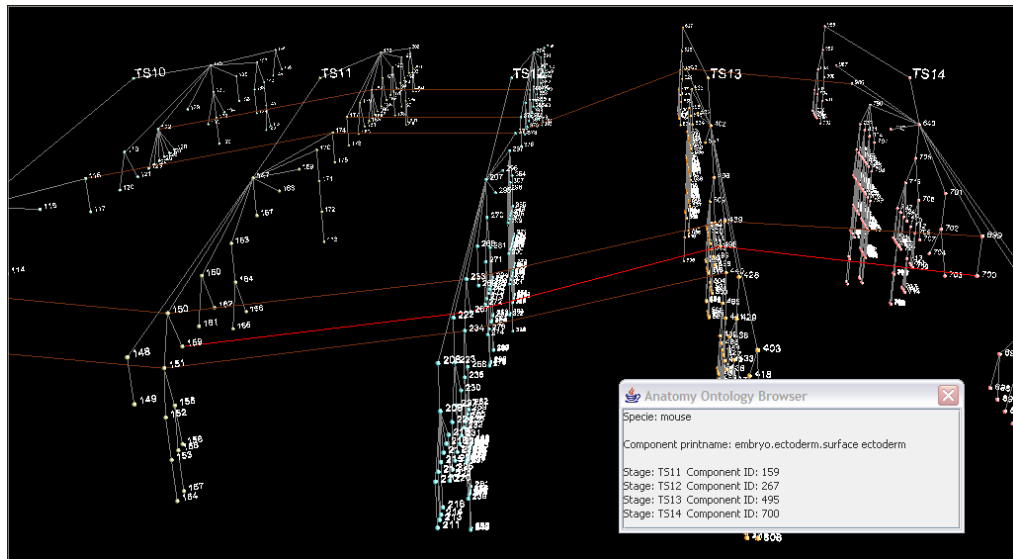


Figure 8.7. The viewpoint is moved above the visual structure to provide an overview of the lineage paths drawn through successive nodes for the five anatomy ontologies loaded in the 3D window. A single trace is highlighted and the IDs of the nodes through which it passes are displayed in a custom dialog.

8.4 Assessment of visualisation solutions

The advantages in a visual overview for developing effective mental models of data structure have been previously discussed. This thesis looks at using node-link graphs to provide overviews of the hierarchically structured ontologies studied. A limitation of this approach is the poor use of space inherent in hierarchical graph visualisation. This contributes to increasing occlusion with data set size, resulting in a reduction in usability of the overviews generated. Data abstraction that hides or fades out data of lower relevance is used to manage the occlusion that occurs, followed by different options for detailed analysis of ROIs, described in detail for the 2D browser in § 6.5.4. An extension to 3D provides additional space for drawing data, presenting further options for resolving limited analysis due to space restrictions in 2D (refer §6.7.5 and §6.7.6).

Research and anecdotal evidence both point to increased perceptual (over cognitive) ability in humans for especially complex data analysis. There are, however, significant variations in human spatial awareness and ability, with an influence on ease of use of especially 3D visualisations, where disorientation during navigation and exploration often occurs. People with high spatial ability quickly identify solutions to problems presented in visual form, easily constructing effective mental maps of data structure that improve ability to decode and retrieve information. Placing markers in data provides cues for orienting users, helping to create maps of data that can be used to build an understanding of its structure. This is achieved in the visualisations generated primarily by using colour to encode data attributes. Annotation in data may be used to point users to ROIs, and provide suggestions for determining relationships within the data and links to external sources with more information, as suggested in [15].

A (second) major usability evaluation of the visualisation browsers was required to assess the potential for improved analysis after the changes made to the browsers. This was to confirm whether solutions had been developed to meet the requirements identified for analysis of the anatomy ontology data being studied, and that could be extended to analysis of other similar data. This evaluation would also look at identifying visual cues that would help to provide intuitive and effective analysis and IR for each target user group. Chapter 9 presents a final evaluation of the browsers, to measure usability of the solutions provided for data analysis. The evaluation was also used to provide answers to outstanding issues for the (visual) analysis required.

Chapter 9

Final evaluation of visual analysis solutions

9.1 Major questions addressed

Evaluations of the visualisation prototypes developed were carried out to determine if any advantages for analysis and IR are provided by the graphical representations of the ontologies over the text indices used in the EMAP browsers. The evaluations were also used to elicit suggestions for methods that would improve analysis currently available using the working EMAP browsers. A number of factors were expected to influence users' responses, including independent factors such as prior use of visual and other types of analysis tools. The metaphors on which the visual structures generated for the prototypes were built and cues available for analysis were also expected to affect usability of the visualisations. It was also important to assess reusability of the browsers; whether it would be possible to extend the functionality provided to analysis of other ontology data.

The analysis of the first set of evaluation results raised a number of issues on the capability of humans for especially complex data analysis, and the influence of perception on effectiveness of analysis. The following points look at specific questions this second structured evaluation was to address, in addition to assessing overall usability of the prototypes developed.

9.1.1 Perceptual cues provided

1. The first (structured) evaluation identified distinct differences in search strategies based on background and domain knowledge.

Biologists (with domain knowledge) made better use of the visual structures - appearing to obtain quickly a good understanding of data structure and easily identifying paths to follow to locate specific components. Good support for navigation through

the data would be especially beneficial for data querying and IR for this target population.

Computer scientists (with little to no prior knowledge of data content) were more reliant on the search dialog, performing what can be described as blind searching. Ability to highlight and locate search matches from within the dialog should be especially useful for this target group. Colour or shape encoding of results based on search criteria and/or relevance of results should also aid IR.

What kinds of (visual) cues could be provided to users that would increase intuitiveness in querying and improve IR? How useful would these cues be to each of the two main (and fairly distinct) target user groups? Would these cues also be useful to other users (falling outside main target) — how can the visualisation browsers be extended for more general use? *Semantic* searching, on synonyms and abbreviations for instance, would serve as an IR aid for all users. Employed with some form of encoding (such as colour or shape) that differentiates search hits from potential matches, this would also allow incremental searching, ranking results based on *degree of match*, useful for retrieving similarity in data especially for empty result sets.

2. Are cues currently provided in the browser useful for IR? Are users able to recognise and correctly interpret these cues? Which of these are useful for which aspects of data analysis? Do those visual cues provided ease identification of (potential) data of interest? Or do they only confirm users' understanding of data? Is it possible to quantify usefulness of perceptual cues provided?
3. Are visual cues alone able to provide useful analysis, especially when searching for specific data, or is (supplementary) textual detail required for correct interpretation of results? How much and what kind of textual information is required for effective use of the visualisations?
4. How meaningful are attributes of the physical objects, such as shape and colour, to users? Are users able to interpret data encoding intuitively and correctly?
Simplicity in encoding was chosen over a large number of options for differentiating data attributes. Does this detract from usability of the visualisations, or do the restricted set of options result in less complex structures that are easier to understand, improving data interpretation? Does the small number of options for encoding limit or improve analysis of ROIs within the context of the overview?
5. How easily are users able to locate specified components based on paths to root alone? How does this compare with locating nodes using the search dialog?
6. Do users remember nodes visited during visual exploration of the data structure? Does ability to locate data of interest increase with repeated searching using the visual structures alone? Are users able to identify and/or place markers in the data that aid location of ROIs and provide the potential for continuous, incremental analysis? What would serve as effective markers to the distinct user groups? Are users able to build a stable

and correct mental model of data structure during exploration and navigation through the data?

7. Does spatial awareness influence ability to navigate through and explore visual representations of data? How does this map to utility of inherently complex 3D visualisations?
8. How well are users able to remember visual structures analysed? Are users able to recall location of objects relative to others? Does spatial awareness map to recall and understanding of semantic content (of data)?

9.1.2 Comparison of the browsers

1. Would it be possible to use 2D alone for the analysis required? What advantages does the 2D browser provide over the 3D?
2. Apart from providing more space in which to display data, what advantages does the 3D browser provide over the 2D?
3. How does functionality provided for detailed analysis of ROIs compensate for high occlusion in the overview? Are the data overviews still useful where occlusion is a significant problem, in isolation and also when compared to use of the text indices?

9.1.3 Additional hypothesis

A final hypothesis was to be tested during this evaluation, in addition to the two presented in § 7.1.1 for the first structured evaluation:

H_{0C} Spatial awareness/ability has no significant influence on use of the visualisation browsers.

H_{1C} Ease of navigation and exploration through especially the 3D browser will map to spatial awareness/ability, with an influence on effectiveness of data analysis and IR.

9.2 Evaluation design

9.2.1 Preparation of evaluation documents

The documents used for this evaluation were based on those for the first structured evaluation (refer § 7.1.2 and appendix C). The user instruction sheet and consent form was modified to reflect differences in the procedure (see § E.1). The background questionnaire administered to users at the start of the evaluation (refer § C.3.1) was not edited. Questions 16 and 17 on the use of the EMAP browsers were, however, added (as part 9) to the post-evaluation questionnaire (see § E.4), to capture changes in use between the two evaluations. The post-evaluation questionnaire, though derived from the first used, has significant differences in parts 4, 6 and 7, with more detailed and focused questions to provide answers for the issues raised in § 9.1.

The task scenario sheets (see § E.3) comprise three main sections, comparing similar tasks performed using the EMAP text indices (shown in figure 5.1), and the 2D and the 3D

browsers developed as part of this project. Additionally, functionality specifically implemented to provide solutions for data analysis and IR requirements for EMAP and XSPAN in each of the visualisation browsers was tested.

Exercise sheets for simple tests of spatial ability/awareness (refer § E.5) were also prepared, the first to test spatial memory, based on the visualisations examined during the evaluation. The last two exercises required users to answer questions based on rotation or orientation of visual structures. It should be noted that the spatial ability exercises are very simple, using sample tests typically employed in psychometric evaluations).

9.2.2 Test run of evaluation procedure

A test run through the task scenarios was performed, to ensure that users would be able to carry out the tasks and evaluate the new functionality implemented. Modifications were made to wording and structure of the task sheets and the post-evaluation questionnaire based on the results of the run, to improve their presentation.

9.3 Implementation of evaluation procedure

9.3.1 User backgrounds

Five users took part in this evaluation: two from MACS and three from the MRC. Four users were classed as biologists, and the last as CS. Two were female and three male, and all users were between 25 and 35 years old. Four out of the five users took part in the first structured evaluation, and the last took part in heuristic evaluations prior to the first structured evaluation.

It should be noted that the user IDs used to report the results of this evaluation do not correspond to those in the first structured evaluation. Because of the restricted number of users beyond reporting some of the results with an indication of research backgrounds no comparison is made between users based on background.

Experience using EMAP browsers

Figure 9.1 illustrates use of the working EMAP browsers for the participants. One user had never made use of the browsers, but all others had used the browsers for over a year, from occasional to daily use.

9.3.2 Evaluation procedure

This followed a similar procedure to that for the previous structured evaluation (refer § 7.3.2). The evaluation process started with a brief explanation of the reasons for carrying out the evaluation, after which each user filled out the consent and instruction form. Only users who had not taken part in the previous structured evaluation filled in the pre-evaluation questionnaire.

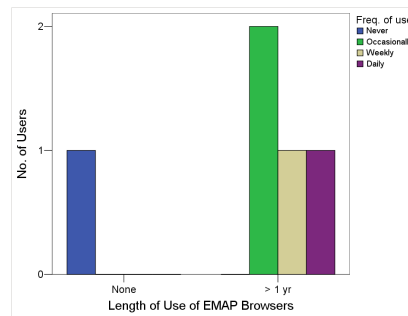


Figure 9.1. Frequency and length of use of the working EMAP browsers

A brief overview of the visualisation browsers was given, and users were provided with a *quick guide* - a single sheet summarising functionality available for analysis using the visualisation browsers (see § E.2). Users then carried out the tasks for the 2D and 3D browsers, recording time to complete each task in seconds using a stop-watch. The visualisation browsers additionally automatically logged and time-stamped functions called.

The exercise testing spatial memory (see exercise 1 in § E.5) was then performed: users were asked to draw their understanding and/or recollection of the structure of the visualisations in 2D and 3D showing the *group* created.

This was followed by the post-evaluation questionnaire and the SUS. The last two exercises measuring spatial awareness (refer § E.5) were then performed. The purpose of these exercises were explained to users: to obtain some measure of their general approach to visual analysis. Users were instructed to skip questions they were unable to answer, and the time to complete each of the two tests was recorded.

A short debrief allowed users to provide any additional feedback and/or obtain more detailed answers to questions asked during the evaluation. Users were then thanked for their participation.

9.4 Analysis of results

Because this is an even smaller group than was used for the previous evaluations the only statistical test performed is the calculation of means within a 95% CI for the SUS and the post-evaluation questionnaires. To allow comparison in one direction all responses to items in the post-evaluation questionnaire were (re)ordered to place all *negative* poles on the left with a value 1, and *positive* on the right with a value 9. Comparison between the text indices and the visualisations placed preference for the indices on the left, at 1 and the visualisations on the right at 9. Comparison between 2D and 3D placed 2D on the left and 3D on the right. The original response sheets were not edited but graphs were drawn to reflect this order.

Conclusions about usability of the browsers in general and usefulness of functions implemented are largely based on user (re)actions recorded during the evaluation and comments made during and at the end of the evaluation process, supported by the quantitative infor-

mation obtained. It is acknowledged that the results are focused on a very small number of (potential) users; a larger number of participants would allow more statistically significant conclusions to be drawn about general usability of the visualisation browsers. However due to restrictions in availability of typical target users, the domain experts and other researchers who work with the bioinformatics data being studied, this analysis will reflect those observations made, to allow a restricted set of conclusions to be drawn that could lead to more focused research on issues of interest brought up as a result of the evaluation.

It should be noted that the spatial ability exercises administered are not sufficient in themselves to provide a complete assessment of users' spatial ability or awareness. The results obtained are only compared with user satisfaction with the visualisations generated and ease of use recorded for the visualisation browsers, to see if there is any correlation between performance in the exercises and use of the visualisation browsers. This information is only expected to provide an indication of spatial ability and awareness; results obtained should point to areas in which further research into spatial ability should focus, in order to obtain more concrete conclusions.

9.4.1 Task completion times

Task times recorded for the main evaluation were disregarded; task completion times for the interactive visualisations are dependent on system response, which may vary significantly with computing resources. With only five users performing the evaluation on three different platforms with large variation in specifications the influence of system response on task completion time was significant. Details on specifications of computers used to carry out the evaluation can be found in § F.1.

Task completion times for the spatial ability exercises were, however, combined with the scores obtained to measure performance.

9.4.2 SUS Scores

The SUS scores are shown in figure 9.2, with a mean score of 65.25, within a 95% CI of 44.04 and 86.52. The highest score recorded was 82.5 and the lowest 37.5.

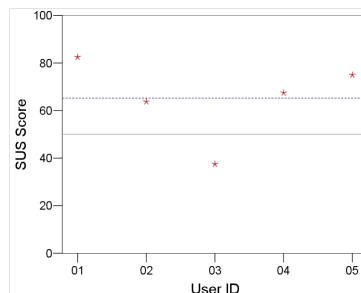


Figure 9.2. SUS scores for participants, with a mean of 65.25.

9.4.3 General satisfaction ratings

Measuring on a Likert scale from 1–9, and disregarding items that were scored as N/A, the overall mean for rankings measuring usability and satisfaction with the browsers was 6.37, with limits for the 95% CI at 5.35 and 7.40. Figure 9.3 shows overall mean satisfaction rating for each user: three lie above the mean and two below. All values fall above the mid-point 5.

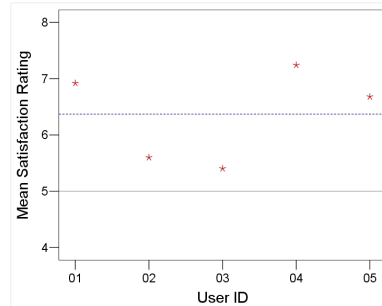


Figure 9.3. Overall mean satisfaction rating for each user for the visualisation browsers

Rankings for items measuring *overall reactions to the system* (part 3 in the post-evaluation questionnaire in § E.4) recorded a mean of 5.417 (with 95% CI between 3.37 and 7.47). The item with the highest ranking is the system *providing adequate power to users* (7.5), followed by *being stimulating* (6.5). The item ranked the lowest was for the *system being frustrating* (3).

Overall means for each of the aspects of usability tested are given below:

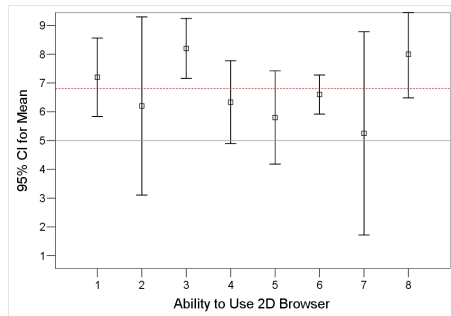
Data Visualisation & Screen:	6.64 [95% CI: 5.71–7.57]
Terminology & System Information:	6.60 [95% CI: 5.25–7.95]
Learning:	6.15 [95% CI: 5.13–7.17]
System Capabilities:	6.41 [95% CI: 3.94–8.88]

Graphs detailing results for individual items on the questionnaire can be found in appendix F. The following sections provide more detail that may be used to answer the questions posed in § 9.1.

9.4.4 Assessment of the 2D browser

Overall mean ranking for ability to use the 2D browser was 6.81, with a 95% CI between 5.57 and 8.04. Figure 9.4 lists the eight items used to measure usability of each of the visualisation browsers, and shows mean ranking for each item for the 2D browser.

Understanding of data structure was ranked highest at 8.20, followed by *system response* at 8.00; this evaluation of the 2D browser worked with only relatively small data sets containing up to 200 nodes (compare with the previous evaluation — refer § 7.4.3 and § 7.5 — where very poor response hampered usability working with data sets containing almost 2000 nodes). *Navigation through the data* was also ranked relatively high at 7.2. *Usefulness of visual cues for querying* recorded the lowest score of 5.25.



1. Navigation through data
2. Location of specific information required
3. Understanding of data structure
4. Understanding of data encoding
5. Querying data for information required
6. Understanding of visual query results
7. Usefulness of visual cues provided for querying
8. System response

Figure 9.4. User rankings for ability to make use of 2D browser, focusing on the eight attributes listed. The broken line shows the mean for all items.

9.4.5 Assessment of the 3D browser

Figure 9.5 illustrates participants' ability to make use of the 3D browser, based on the items listed in figure 9.4. Overall mean ranking for the 3D browser was 5.71, with 95% CI between 4.41 and 7.02. Larger variation is recorded in user responses than for the 2D browser.

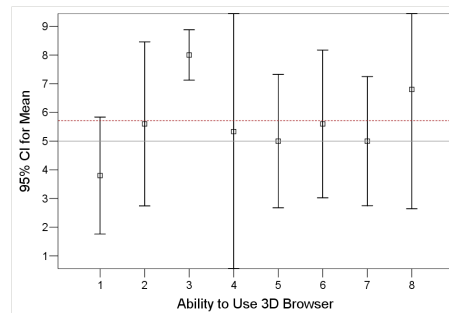


Figure 9.5. User rankings for ability to make use of 3D browser, based on the list in figure 9.4

As for the 2D browser, *understanding of data structure* and *system response* scored the two highest rankings at 8.00 and 6.80 respectively. One item fell below the mid-point — *navigation through the data*, at 3.80. *Querying of data for information required* and *usefulness of visual cues for querying* both fell on the mid-point (the two lowest rankings for the 2D browser).

9.4.6 Comparison between the 2D and 3D browsers

Although there were acknowledged advantages in the 3D browser the 2D scored higher overall rankings for learning and actual ability to use (refer § 9.4.4 and § 9.4.5). The main reason for this, from user observation and comments made, is labelling of data; due to memory limitations in Java3D (discussed in § 9.6.1) only component IDs are used to label nodes in 3D. This results in a higher cognitive burden on users as more effort is required to remember which ID represents a specific node — one user commented that not *thinking in numbers* made the labels difficult to relate to. The visual structures are unable to provide enough information on their own to determine with certainty which anatomical components specific nodes represent; a second step was often required to retrieve further textual detail (using the *component detail* dialog).

User tasks required locating specific nodes visually so that inherent complexity in 3D navigation was also a significant factor in assessing usability. For two users poor system response due to limited computing resources made navigation in the 3D browsers especially difficult, further reducing usability.

Figure 9.6 compares mean rankings for ability to make use of the two visualisation browsers based on the items listed in figure 9.4, detailing also, user backgrounds.

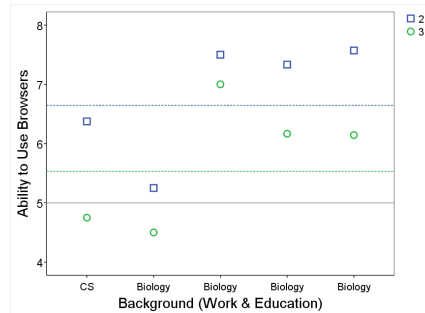


Figure 9.6. Comparison of ability to use 2D and 3D browsers, based on the items listed in figure 9.4. The chart also indicates each user’s background. (Note that order of users on the categorical axis does not follow order of user IDs in other figures.)

The main challenges in 2D are restrictions in space and the poor use of screen real estate common to hierarchical graphs, limiting the amount of data that can be displayed effectively in each 2D window. There are, however, distinct advantages in 2D, made apparent in this evaluation; the simpler representation of data structures makes the 2D graphs useful for analysis of individual data sets. Ability to trace paths easily within the data was found to be especially useful in locating specific nodes. Measuring actual ability to make use of each browser users recorded higher ability to recognise functionality available in 2D and make use of it for the analysis required.

However the approach used is not able to provide solutions for all user requirements using 2D alone. The extra space available in 3D is necessary to visualise multiple data sets simultaneously, to allow direct comparison between data and provide graphical support for the identification and display of relationships that cross data sets. This is confirmed with a direct comparison between the 2D and 3D browsers, which shows a leaning towards preference for use of the 3D browser, illustrated in figure 9.7: the mean, represented by the broken line, lies at 5.60, with 95% CI between 4.62 and 6.58. (Numbering for items on the vertical axis correspond to those in the post-evaluation questionnaire which can be found in § E.4, described also in table 9.1).

Obvious advantages in 3D include *tracing of lineage* across stages of development (refer figures 8.7 and 9.9), which recorded the highest mark at 8.4. Users also indicated a preference for *grouping* in 3D — the next highest score was 7.3, for *usefulness for highlighting groups*, and *support for creating and displaying groups* had a mean of 6.8. (A graphical comparison of the display of a user-created *group* in the two visualisation browsers can be found in

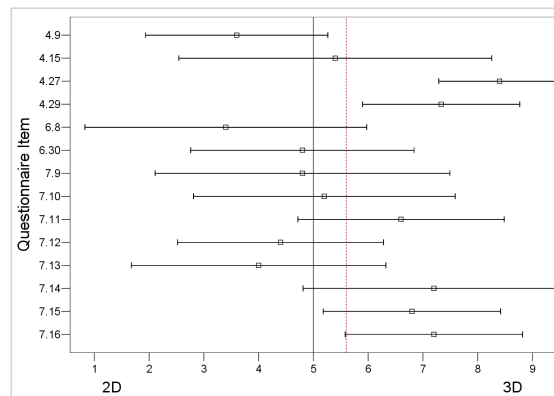


Figure 9.7. Comparison of the 2D to the 3D browser shows a preference for the 3D; the mean shown by the broken line is 5.60, measured along the Likert scale from 1–9. (Items used to compare the browsers are listed in table 9.1.)

Table 9.1. List of items used to compare 2D and 3D browsers

Item	Rank
4.9 Ease following ordering of components	3.6
4.15 Intuitiveness of navigation through the data structures	5.4
4.27 Usefulness for tracing lineage	8.4
4.29 Usefulness for highlighting groups	7.3
6.8 Ease of learning of the functions available	3.4
6.30 Querying	4.8
7.9 Navigation through visual structures	4.8
7.10 Data analysis	5.2
7.11 Ability of visualisations to provide an overview of data structure	6.6
7.12 Usefulness of visual structures for analysis	4.4
7.13 Locating data of interest	4.0
7.14 Identifying relationships in data	7.2
7.15 Support for creating and displaying groups	6.8
7.16 Functionality for tracing lineage	7.2

figure 6.29).

Users found *ordering of components* easier to follow in 2D, with a ranking of 3.6, again explained by the provision of more meaningful labels. The 2D browser was also found to be more intuitive when it came to *learning how to use functions* at 3.8. *Locating data of interest* was the next item found more useful in 2D, with a mean ranking of 4.

9.4.7 Comparison of the visualisation browsers to the EMAP indices

Figures 9.8 and 9.9 compare use of the EMAP text indices to the visualisation browsers. The chart indicates higher usability of the visual structures for the tasks carried out, with a mean of 6.73 within a 95% CI of 5.87 and 7.58.

All users found the visualisations easier to use than the text indices for *tracing lineage*, giving it the highest score possible of 9. *Grouping of data* was the next highest with a mean score of 8.4, followed by *ease of use of the data structure* with a mean of 8. User comments recorded in the post-evaluation questionnaire also recommend the visualisation browsers as a solution to requirements for graphical support for *grouping* in EMAP. The only item for which the text indices were found easier to use was *search and query*, with a mean of 4.4.

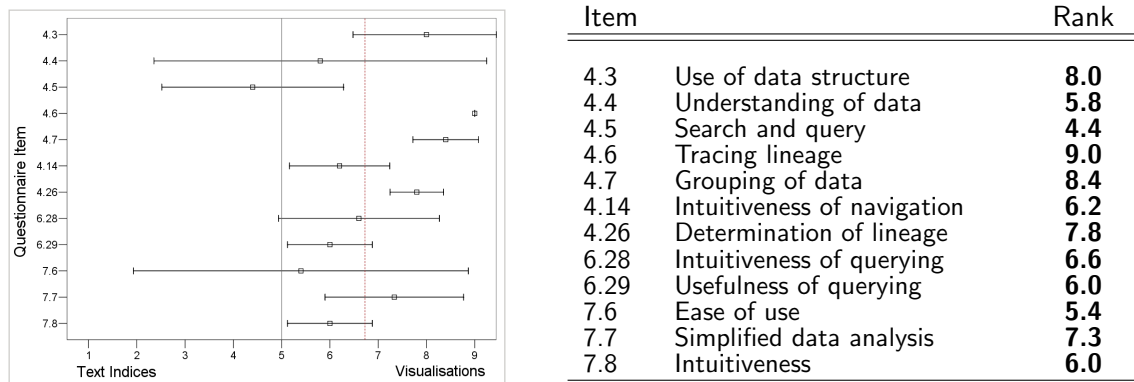


Figure 9.8. Comparison of the EMAP text indices to the visualisation browsers shows higher usability of the visualisations for the analysis required. (Numbering for items on the vertical axis correspond to those in the post-evaluation questionnaire which can be found in § E.4, described also in the table on the right.)

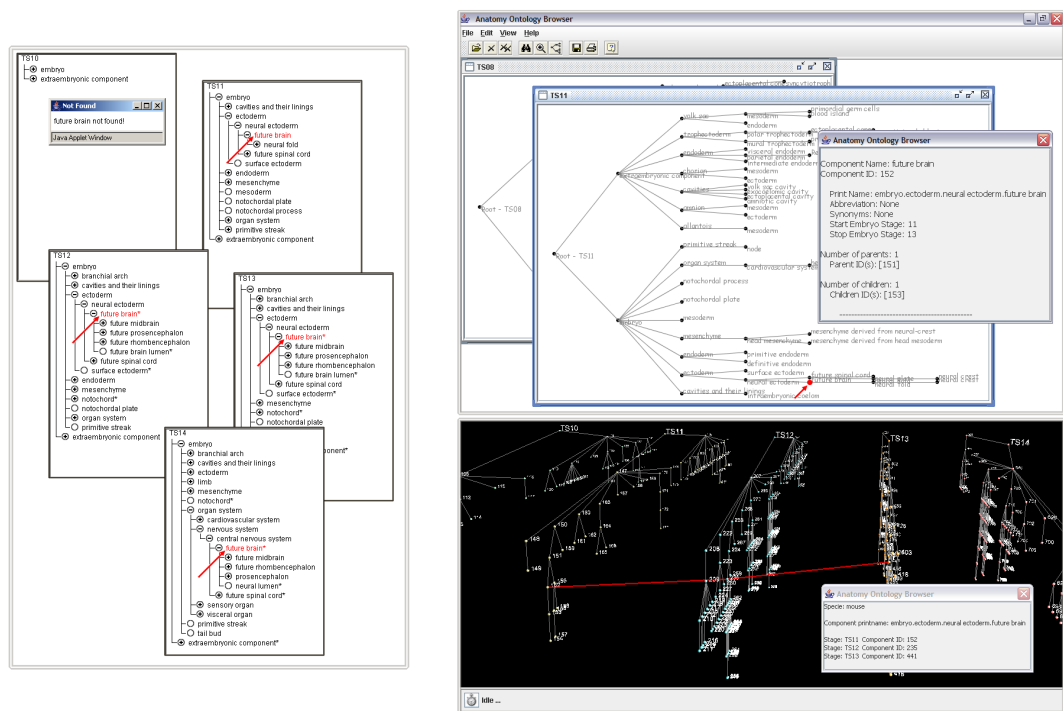


Figure 9.9. The image shows the steps required to carry out T3-2D and T2-3D on the Task Scenario Sheets, to determine the stages through which the component *neural ectoderm.future brain* persists. Multiple stages must be searched in the text indices to solve this problem. The same applies for the 2D browser, but this has the advantage that supplementary textual detail will reveal this information once the first instance of the component is found. The 3D browser provides the most intuitive and informative solution to this problem, by providing a physical trace showing lineage across multiple stages.

9.4.8 Spatial ability exercises

Exercise 1

The first exercise asked participants to use a simple drawing to show their understanding and/or recollection of creating a *group* in 2D and viewing the same *group* in 3D. This was used to test memorability of the visual structures, and also determine if users recognised any advantages in display of *groups* in 3D. Figures 9.10, 9.11 and 9.12 show a sample of the results obtained.

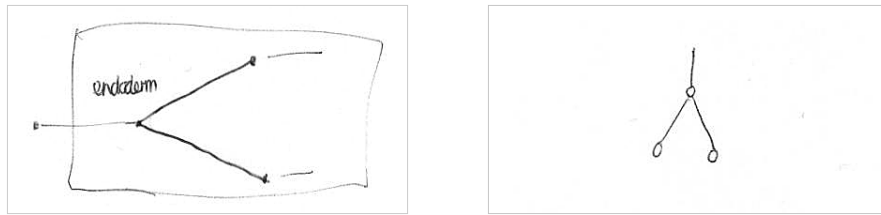


Figure 9.10. The 2D structure on the left, drawn using the default LR layout, contains a label, indicating the importance of labels in the graph. The 3D representation on the right however makes no use of labels, highlighting the difficulty users had in interpreting labels in 3D. The 3D structure is drawn using the TD layout, the only option available in 3D.

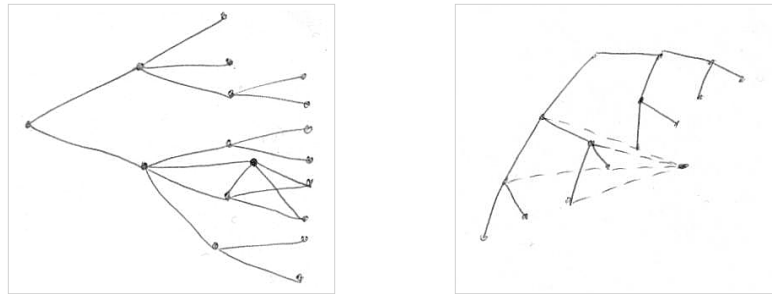


Figure 9.11. The 2D structure on the left shows the crossing of nodes that occurs when the *group* is created. The drawing on the right shows the advantage in removing the *group* created to a plane parallel to that holding the DAG. The impression of perspective can also be seen in the 3D structure.

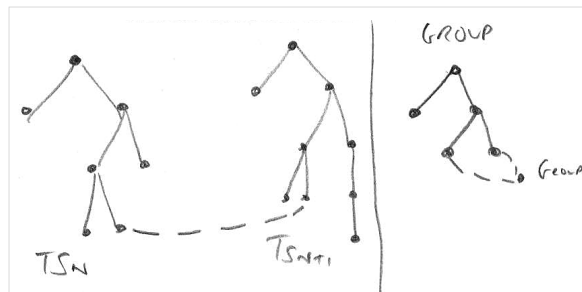


Figure 9.12. The 3D structure, using a broken line to show continuity in the visualisation. The smaller section on the right also uses a broken line to show the links drawn to the group node from the main DAG.

Exercises 2 & 3

Exercise 2 tested spatial orientation of objects, and exercise 3 required users to determine the next in a sequence that transformed, rotated and/or translated objects based on a logical pattern. (Instructions for all three exercises can be found in § E.5.)

Three users answered all questions correctly and two answered one out of nine incorrectly in exercise 2. One user answered all questions correctly in exercise 3, three answered 1 and the last user 3 questions incorrectly. Time to complete each exercise varied from more than 2 to almost 6 minutes. The results obtained are presented with a description of the results of the memory exercise in table 9.2.

Table 9.2. Results for the spatial awareness / ability exercises. Scores are recorded as percentage of number of correct questions out of the total posed, and completion time is shown in minutes.

ID	Memory exercise		Exercise 2		Exercise 3	
	2D	3D	Score (%)	Time (min)	Score (%)	Time (min)
01	detailed representation of DAG in 2D showing <i>group</i> and highlighting crossing of links that resulted	vertical layout illustrating effect of perspective projection; <i>group</i> drawn extending out from main graph	100	2:19	87.5	2:17
02	distinct DAGS drawn using vertical layout; <i>groups</i> drawn separately using broken lines to show links to <i>group</i> node	drawn as for 2D but with broken line showing continuity between DAGs; <i>group</i> drawn as for 2D	100	5:51	87.5	3:56
03	horizontal layout drawn, labelling provided but <i>group</i> not shown	vertical layout drawn <i>without</i> labels, no <i>group</i> shown (user was not able to view group in 3D)	88.9	3:25	100	5:00
04	graph drawn with horizontal orientation, using broken lines to show connections between <i>group</i>	- (user was not able to view group in 3D)	100	4:02	87.5	4:35
05	- (user unable to draw visual structures from memory)	-	88.9	2:19	62.5	2:33

9.5 Discussion of evaluation findings

9.5.1 Understanding of data structure

Users generally recorded obtaining a good *understanding of data structure* using the visualisation browsers, confirming findings in research of one of the main advantages in visualising data. This was reinforced by high usability recorded for tracing lineage (refer figure 8.7) and creation and display of groups using the visual structures (refer figure 6.29); the visualisations provided a clear advantage in graphical support for both options.

The tasks carried out mostly required users to search through the text indices and

the visual structures to locate specific nodes. Good support for what is also inherently simpler navigation in 2D aided visual exploration of the data, allowing intuitive formation of mental models of data structure; users commented on usefulness of functionality that allows highlighting of paths in a single tree in the 2D browser. Semantically meaningful labels in 2D also served as a navigation aid; users were better able to remember paths previously followed, allowing data of interest in related data sets to be located quickly and easily. Navigation through the data structures in the 3D browsers however recorded low rankings — inherent difficulty in navigation in 3D coupled with labels with low semantic relationship to users' understanding of data (for those users with domain knowledge) increased difficulty completing tasks involving (visual) search and query.

Practice is required to be able to navigate effectively in 3D space using a three-button mouse as was done for this evaluation, and users were generally observed to improve their control over navigation as the evaluation proceeded, where the resources available did not hamper system response. Stronger perceptual cues are, however, still required to improve ability to locate data of interest.

9.5.2 Search and query

Users found searching for specific information difficult in the visualisation browsers. The search dialog obscures a large portion of the window, sometimes hiding search hits highlighted in the graph, so that users lost the supporting visual information. Moving the dialog away in order to see the visual query results meant losing the corresponding text listing *component IDs* and *print names* for search hits. Further, very long *print names* meant most users widened the search dialog to read results without having to scroll, covering even more of the visual structures.

One difference between text searching in 2D and 3D is that each search in 2D is performed (independently) only for the graph with the current focus; it is necessary to open a separate search window for each graph. In 3D, however, searching is on all visible nodes in the window. Search hits are only labelled by *component ID* and *print name*; also providing tree labels would aid location of hits in the graphs in 3D.

Visual search in the 3D window is further complicated because the node labels showing *component IDs* are not easily translated into meaningful symbols for users; it is necessary to look up the component names that match the IDs displayed. This underscores the importance of supporting semantically meaningful information for effective data analysis and IR. This may have contributed to users indicating a preference for *search and query* using the text indices, and the low scores recorded for querying the data and locating specific information in the visualisation browsers.

Nodes in trees are ordered by *component ID*, to provide uniformity in layout for each data set and across related data sets. Because users did not find use of the *component IDs*

intuitive this ordering, which should be obvious in 3D, was not recognised, nor was it found to be more useful than labelling using *component names* as the default in 2D. Providing the additional option for ordering by *component name* should improve location of data of interest.

The role that intuitive navigation plays in successful searching was made apparent by the difficulty users had locating specific information in the 3D browser. The ability to *jump* directly to a node of interest in 3D would be useful especially where high density of data occurs; most users asked if this option existed. A solution which would remove the disorientation, that sudden changes in location in virtual worlds often leads to, would be gradual animation that also allows users to explore data structure as they follow paths through data to arrive at specific objects of interest. This would also provide the equivalent of tracing paths through single graphs in 2D.

9.5.3 Managing occlusion

The user evaluations, performed for the prototypes developed, confirmed the advantage in a visual overview of the anatomy ontologies under study. However, occlusion presents a problem visualising data sets containing nodes beyond a fairly low threshold, as discussed in the design of the visualisation browsers in chapter 6; a recognised problem in the use of hierarchical graphs is poor scalability.

Options for managing occlusion focus mainly on data abstraction, displaying only a user-specified number of levels in the graph and ghosting out data (in 2D), folding away sub-trees with lower interest in both browsers and highlighting data of higher importance. Additional options include extraction of ROIs to sub-windows for analysis in isolation. Figure 9.13 shows mean rankings for usefulness of each of the options listed for managing occlusion in the 2D and 3D browsers at 8.16, with 95% CI between 7.44 and 8.86.

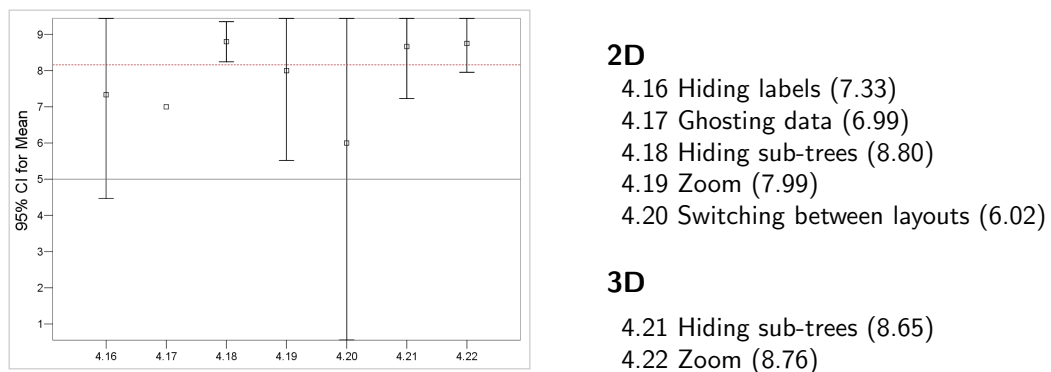


Figure 9.13. Measurements of usefulness of options provided for detailed analysis of ROIs. Means are given over all users for each item.

Even though users found that the options provided were useful for analysis in ROIs, this was not reflected in use of the overview where significant occlusion occurred; this item (4.12 in the post-evaluation questionnaire) was ranked at 5.2, within a 95% CI of 2.52 and

7.91. A solution being considered to this problem is to collapse sub-trees in areas of very high density into composite nodes, to obtain an effect similar to that shown in figure 2.12 and described in [98]. Functionality implemented for analysis of ROIs may then be used to retrieve detail for each composite node.

9.5.4 Perceptual cues

One of the main issues this evaluation sought to address was the identification of effective perceptual cues for especially search and query, to provide intuitive IR. Visual cues provided include highlighting hits in the graphs during text-based searches, with the option to expand searching to synonyms and abbreviations used to describe nodes. The latter is especially useful for instances where search result lists are empty.

Markers may also be placed in the data by attaching comments to nodes and links; this, however, provides only additional textual information, without a visual component. A text-based search may, however, be used to highlight all nodes that have been annotated. Providing the option to place the equivalent of a physical *flag* along or on data objects may aid returning to specific objects.

Visual cues and encoding provided in the visualisation browsers were difficult to recognise or make use of in isolation; supporting, semantically meaningful textual information played a large role in confirming results of visual searching. This was especially apparent in 3D where users required a large amount of effort to locate specific nodes; the labels displaying only *component IDs* did not appear to satisfy users even where they matched those listed in the search dialog. This meant users often went on to retrieve more detailed textual information to confirm search results. Among user suggestions for a solution to this problem were to pop up the *component name* for the node with the focus, a solution that was being considered prior to the evaluation. This would require lower user effort to display *component names* and take up less screen space than the component detail dialog. It would also provide a significant advantage during exploratory navigation as additional information on nodes would be uncovered as users move through the information space, without increasing complexity and occlusion.

Not all users immediately recognised highlighting of search hits (in green); as discussed in § 9.5.2 this may have been because the graphs were often partially obscured by the search dialog. Once its significance was understood, however, confidence in visual search results appeared to improve, with users locating nodes required more quickly. Encoding results using a change in shape or size as well as colour may help to provide stronger perceptual cues, especially for areas with high density of data. The option to *fade out* non-search hits, suggested in the previous evaluation, would also help to identify potential nodes of interest by lowering the distraction of other objects in the scene.

One user only took advantage of the option to reset colouring of nodes and links in the 3D

browser, changing the default colour for two DAGs loaded into the window to provide greater contrast between them. The colour codes were also changed for a set of mappings, to provide greater contrast between the links drawn and the background. Variation in colour coding of nodes in a single graph may provide useful options to users for visually annotating data, creating *markers* that highlight relationships in the data and aid IR. This should also help to resolve the problems revealed in the first structured evaluation that reduced usability for a colour-blind user, where specific colour combinations prevented correct decoding of data attributes and increased difficulty recognising objects highlighted in the graphs.

9.5.5 Spatial awareness/ability

Results from the spatial memory tests showed, for the four users who completed it, a fairly good understanding of data structure. One user displayed very good recollection of the structures in both layouts, showing a large amount of detail and clearly distinguishing display of the *group* created in 2D and 3D. It can be concluded with a good degree of certainty, also confirmed by user comments and responses to the post-evaluation questionnaire, that the visualisations allow users to obtain a good understanding of data structure.

The main difference in results for the additional spatial ability tests was time to complete each exercise. The participant with the most detailed and accurate representation of the visual structures also recorded the shortest times for both exercises, the highest score for the SUS and the second highest mean rating for satisfaction with the visualisation browsers. Although, following the trend for all other users, this user had difficulty managing navigation in the 3D browser, this was observed to improve as the evaluation progressed. This user's responses to the post-evaluation questionnaire recorded a preference for the 3D browser, finding it more intuitive even for navigation, for providing a good overview of data structure, and aiding the identification of relationships within the data.

The user with the longest overall times for completing the exercises provided a fairly detailed drawing of data structure but showed little distinction between the layout in the two views. Although difficulty was recorded for navigation in the visual structures this user still found the visualisations to be more intuitive than the text indices. Navigation in 2D was, however, found to be much easier than in 3D; this user found tracing of paths through the 2D graphs fairly easy, recording that it aided location of data in 2D. In a direct comparison between the two views this user, however, recorded more intuitive navigation and easier recognition of relationships using the 3D browser.

The user who recorded the second longest completion times for both exercises was observed to handle navigation in the 3D browser most easily, obtaining good (over)views of the data structure. This user recorded increasing confidence in navigation in 3D with time. However, navigation in 2D was recorded to be more intuitive, and identification of relationships in the data was also found easier in 2D. Following the general trend the 3D visualisations were found to provide a better overview of data structure. Having unloaded

all data sets before making the switch to the 3D browser this user was unable to observe the group created in 3D, so that a comparison could not be made between understanding of the visual structures in the two browsers. This user recorded the highest overall satisfaction rating and the third highest SUS score.

The user with the lowest SUS score and mean satisfaction rating recorded the third longest overall times for completing both exercises. Although the structures drawn did not show a large amount of detail this was the only user who labelled nodes to indicate relevance of labels in 2D. User comments indicated the 3D representation was not labelled as the labels for this browser were not found to be very useful. Responses to the post-evaluation questionnaire recorded finding the text indices, used with the (pictorial) 2D slices of the embryos, providing easier understanding of data. Navigation through the data in the text indices was preferred to the visualisations, and also, navigation in 3D was found to be more difficult and less intuitive than in 2D. Location of data based on path to root in 2D was found to be easy. The 3D visualisations were, however, found to provide a better overview of data structure than the 2D. It should be noted that this was one of the two users who experienced very poor system response using the 3D browser, and also was unable to view the *group* created in 3D.

One user answered two more questions incorrectly than did all other users and was unable to recall the structures obtained for the *group* created. However, this user recorded the second shortest times for completing the exercises and relatively high scores for the SUS and overall mean satisfaction with the visualisation browsers. This user made extensive use of visual scanning, reorienting DAGs in the 3D browser to more closely approach and examine nodes of interest. The component detail dialog was however repeatedly displayed to confirm identity of nodes in 3D, indicating again difficulty relating to the labels displaying only *component IDs*. Navigation through the visual structures was found to be difficult; however they were still found to be more intuitive than the text indices, with 2D recorded as being more intuitive than 3D. This user found it easier to locate data of interest in 2D, but found identification of relationships within the data using the 3D browser to be easier. This was the only user who found the 2D visualisations to provide as good an overview of data structure as the 3D.

The first user described performed all three spatial ability exercises exceptionally well, and this was found to correspond to overall satisfaction recorded for use of the visualisation browsers developed. Ability shown in use of the browsers also increased with time for this user. However one other user who did not perform as well in the spatial exercises showed more intuitive performance in navigation, but recorded lower satisfaction using the 3D browser. The user with the lowest performance made the most extensive use of visual searching through the data. Outside the two extremes the results are mixed - difficulty in mapping text search results to the visualisations in especially 3D hindered location of nodes, with a significant effect on usability recorded for searching. Further testing and research are

required to provide stronger indications of spatial ability, and determine exactly how strong the correlation is between spatial ability and use of especially the 3D browser.

9.6 Review of visualisation browsers

The visualisation browsers develop novel techniques that provide solutions to the specific data analysis requirements identified, employing an interactive visual data analysis solution. The aim is to provide intuitive methods that allow users to become immersed in and explore individual or multiple data sets simultaneously, to increase understanding of data structure and aid identification of relationships in data.

This and previous evaluations have been used to examine usability of the application developed and to ensure that functionality implemented provides solutions that meet users' information requirements. The information obtained from the evaluations was used to identify further modifications required to functionality developed for analysis. Major changes necessary, described in the following sub-sections, are improvements to functionality for searching through the data and support for navigation in 3D.

2D browser

Options for searching within the graph with the focus work effectively for the data set of interest. It is often necessary, however, to extend searching to retrieve information in other related data sets; additional options are required to widen search to include all data sets loaded in the browser, and also retrieve relevant information from additional (specified) data sets not loaded in the window. The option to collapse/hide the search dialog without clearing visual results would also help users to identify search hits in the graphs more easily.

3D browser

As for the 2D browser, options for searching on additional data sets not loaded in the window are necessary. To aid location of specific nodes the result list should also provide information on the tree to which each hit belongs.

Options for animation are being considered to aid users in mapping paths to nodes of interest; especially where there is high density of data, locating specific nodes is hampered further by inherently complex navigation in 3D.

Functionality for creating, editing and displaying links across trees needs to be made more intuitive; labels on the *mappings* sub-menu require editing to clarify their functions. Options for querying the data also need to be extended to allow greater distinction between the different types of mappings that exist.

9.6.1 Limitations of approach

Reasons for the choice of visualisation techniques to build on and programming language for development were discussed in § 5.5 and 6.1 respectively, looking at the advantages each of these provides for analysis of the anatomy ontology data being studied. Limitations associated with each option are discussed in the following sub-sections.

Hierarchical graphs

Although node-link graphs provide an intuitive visualisation option for hierarchically structured data, they suffer from poor scalability, making sub-optimal use of screen space for laying out data. This was managed by providing specialised functionality for detailed analysis of ROIs, both in isolation and within the context of the overview, the latter being the preferred option. Further investigation of perceptual cues is required to identify additional options to improve intuition in analysis of ROIs.

Performance in Java

Three main reasons contributed to the choice of Java for development: to provide a similar interface to current tools in EMAP and to provide cross-platform and potentially, web access to the visualisation solutions developed. Java's cross-platform compatibility unfortunately comes with a cost in performance due to the extra overhead incurred in the interpretation of the Java bytecode into native machine code, manifested in the 2D prototype by the large increase in system response time observed as data load increases. This is compounded by a significant decrease in program execution speed for remote execution in X-Windows, when compared to MS Windows; enhancements for Swing in Windows have the reverse effect in X-Windows. § 10.4.2 discusses potential solutions for improved system response.

Memory management vs interactivity in Java3D

Requirements for interactive analysis include the ability to select objects of interest in each graph, to allow further information to be retrieved and/or apply encoding to data objects based on the values of attributes. Especially where a large number of similarly structured objects are being built in a Java3D scene, reuse of geometry is the ideal option, to minimise resources required to create objects. However, this limits options for interactivity, allowing objects to be *picked* based only on their bounds. Density in data in the larger graphs, however, means that bounds of nodes and links often overlap, rendering picking inaccurate. The solution to this is to create independent geometry for each object so that picking is able to use more highly defined geometry, with a significant increase in accuracy in selection of objects of interest, but with a corresponding increase in especially memory required to draw data objects. The Java Virtual Machine (JVM) therefore quickly runs out of memory for multiple loading and unloading of especially the larger data sets.

Rotation of the visual structures in 3D also results in rotation of labels away from the viewpoint; it is necessary to orient labels so that they always face the viewer and can be read. This is achieved using the Java3D **OrientedShape3D** object, built from geometries obtained from individual **Text3D** objects created for each node label. Memory required to build each of these objects increases with the number of nodes drawn to the 3D window. To manage memory use labels drawn for a single tree are set to a limit between 200 and 300, starting from the root, so that nodes in lower levels in large trees are not labelled. Textual detail for all nodes and user-created links may still be brought up by double-clicking on objects of interest.

Further, limited support for the development of as well as development using Java3D may result in limited functionality in applications built, and that might not have support in the future. Also, not being part of the standard JRE, users must install Java3D as well as any non-standard Java3D extensions in order to make use of applications developed using these libraries, an added burden for users.

9.7 Summary

This chapter presented a set of questions that analysed the information requirements of target users, looking at issues that require resolution in order to provide intuitive solutions for the challenges that occur in analysis of ontologies in biology, with a focus on anatomy ontologies.

A final, structured evaluation was carried out with a small number of target users, to obtain information required to resolve these issues, and also examine usability of the functionality provided for analysis. An examination of the evaluation results provided further information on usability and utility of the applications developed, identifying functionality that needs to be developed further to satisfy users' information requirements more fully. A review of the questions posed at the beginning of the chapter found that more focused research is necessary to identify additional cues for visual analysis that harness more effectively human perceptual and spatial ability for intuitive analysis.

The chapter concluded with a summary of the modifications required to functionality provided in the visualisation browsers, then addressed limitations of the approach used to develop the visual analysis solution. The next chapter summarises the work done in this project, discussing the main findings in this thesis and the contribution it makes to research. The thesis concludes with directions for additional research that could lead to further options and improvements for visual analysis.

Chapter 10

Conclusions

10.1 Review of thesis

This thesis reviews research in information visualisation, examining methods available for visual data analysis. The aim of this project was to develop visual solutions for analysis of (hierarchically structured) anatomy ontology data that reveal similarity between components in different data sets, with the potential to extend solutions developed to analysis of other similarly structured data.

A large number of techniques exist for generating visualisations, varying between broad solutions that provide general analysis of different types of data and specialised applications providing detailed analysis restricted to specific data types and/or formats, often applicable to only a small number of fields. The aim of each of these applications is to harness human spatial and perceptual ability and reduce cognitive load in data analysis, helping users to form effective mental models of data structure that aid the retrieval of information stored within data.

Challenges for effective visual analysis include the generation of visualisations that are not just aesthetically appealing, but that are able to highlight patterns and relationships within data and provide effective IR. This requires a good understanding of typical users' backgrounds and domain knowledge, familiarity with data analysis tools, and, more difficult to obtain, measures of users' perceptual ability and spatial awareness.

Chapter 2 reviews previous research and current work in information visualisation, leading to chapter 3 which focuses on graph visualisation, the preferred solution for the analysis of the hierarchically structured ontologies. Chapter 4 continues to look at the analysis of bioinformatics data, with a focus on anatomy ontologies and previous research into the analysis of data in the cross-disciplinary field.

10.1.1 Identification of problem area

For practical reasons this thesis restricts evaluation of solutions developed to analysis of a sub-set of the data that contributes to research in bioinformatics, to allow usable options

for analysis to be developed that could then be extended to wider areas of application. Chapter 5 describes user information requirements typical to research that makes use of anatomy ontologies, identifying gaps in existing data analysis solutions. Areas for which improved solutions for analysis are required include:

- the identification of equivalence and other relationships defined between data elements
- automated retrieval and display of temporal relationships across multiple data sets.
- the need for graphical support for the creation of alternative (user-defined) relationships and structures within the data sets of interest

Partial solutions at best were found (in existing tools) for the analysis required (refer § 5.4.1); the project therefore continued to develop alternatives to satisfy the requirements identified.

10.1.2 Development of a visual analysis solution

An interactive visual solution was developed that presents first an overview of the data sets being analysed, to reveal data structure and allow users to obtain an understanding of the information each data set contains. Chapter 6 describes the visualisation browsers created, building first on existing methods, to allow an assessment of techniques currently available for graphical analysis. The chapter then describes techniques implemented for detailed analysis of ROIs in 2D (refer § 6.5.4), and identification and visualisation of relationships across multiple data sets, employing an extension to 3D (refer § 6.7.5).

An assessment of the solutions developed provided a measure of the advantages gained analysing data in visual form over presentation in text; a series of heuristic evaluations and the structured usability evaluation detailed in chapter 7 provided information on the advantages users perceived in visual analysis. The most important of these are enhanced ability to obtain an overview of data structure and more intuitive identification of relationships between data elements. The number of participants in the structured evaluation is too small to draw conclusions based on (strong) statistical significance; however qualitative information obtained from user observation during the evaluation confirmed trends in responses to post-evaluation questionnaires that suggest that visual analysis does provide advantages over textual analysis of the data. This conclusion is also supported by research in the field that shows improved capability for analysis that harnesses advanced human perception.

Analysis of the evaluation results also brought up a number of questions on spatial awareness in humans and identification of techniques and/or cues that harness perception and spatial ability for intuitive analysis. Visual solutions for the specific analysis required were developed further, based on the information obtained from the (heuristic and structured) user evaluations performed. More focused research into human perception and the influence of spatial ability and awareness on perceived usability of visualisation applications was performed, looking for additional insight into the development of effective, usable visual analysis techniques.

Chapter 8 summarises changes made to the prototypes and further work on the tech-

niques developed, to increase intuitiveness in the analysis solutions to more fully satisfy user requirements. Chapter 9 presents the questions and other issues in visual analysis brought up during the course of this project, that were to guide a final evaluation, assessing the benefits of visual analysis, advantages of each of 2D and 3D for generating visualisations, and the influence of human spatial ability on visual data analysis. Analysis of the evaluation results provided information for answering some of the questions raised, and point to further research directions that may provide stronger conclusions on what remain open questions in human spatial and perceptual ability and their influence on visual analysis.

10.2 Main findings

This thesis studies information visualisation, to determine its applicability to complex data analysis and its ability to improve analysis. It provides confirmation that visual overviews improve understanding of data structure [163, 179], providing a powerful alternative to traditional textual analysis. Contrary to research on the influence of spatial awareness on (perceived) usability and analysis capability, user backgrounds and domain knowledge were found to have the most significant influence on strategies used for querying data and locating information of interest, with no noticeable influence due to gender, especially in the absence of or failure to identify (visual) markers provided to manage disorientation and aid exploratory navigation.

It was also discovered that despite improved ability to obtain an understanding of data structure and retrieve information of interest using visual representations of data the absence of supporting and semantically meaningful textual information significantly degraded confidence in results of analysis. The need for perceptual cues that users recognise and understand is critical in the development of intuitive visual analysis solutions. Identifying cues that are semantically meaningful to the different target users with varying research backgrounds common to cross-disciplinary fields such as bioinformatics is a challenge that requires detailed study of the interaction between humans and computer-based systems as well as a good understanding of the unique information requirements of the different research fields that contribute knowledge to analysis of data such as the anatomy ontologies this thesis studies.

Closely working with the target population for which solutions to data analysis were being sought highlighted issues that limit current solutions for visual analysis: wide variation in human perceptual and spatial ability and methods used for querying and IR limit the reusability of solutions developed; a choice has to be made between building general tools that provide only simple analysis and specialised methods for focused data analysis solutions. This thesis starts by looking at a general solution but leans toward the latter; it, however, employs modular development that should aid extension of the solutions developed for wider analysis.

10.2.1 Contribution to research

As part of the research performed for this thesis two visualisation prototypes were built, first to evaluate the potential of existing solutions for the specific analysis required for this project and to provide a tool which allows multiple techniques to be used in concert for visual data analysis. Based on learning from heuristic evaluations performed with a sample from the target population and a review of user requirements further functionality was implemented, focusing on the development of techniques for detailed analysis of ROIs and management of occlusion in individual data sets.

Functionality was then developed to aid the identification and display of relationships across multiple, related data sets. A novel contribution to existing techniques for visual analysis of hierarchical data is described in § 6.7.2; the simple node-link graphs drawn in 2D planes to provide overviews of individual data sets are layered in parallel along the horizontal axis in 3D space, allowing simultaneous display of different ontologies. The space between DAGs is used to hold the relationships that cross multiple data sets, employing colour-coded links drawn between node pairs. This provides a method for visualising temporal relationships in the data such as lineage during stages of development in a specified organism. Other types of relationships such as equivalence between corresponding components in different organisms may likewise be drawn between node pairs across DAGs.

This is extended in § 8.3.2, which describes an alternative to encoding relationships using links between nodes pairs: colour-coded wireframes drawn round each node to denote those with (user-defined) relationships with other nodes. This serves two purposes: it reduces occlusion in the 3D window, and more importantly, allows relationships to be displayed even where only one of a pair of related nodes is drawn to the screen.

Graphical support is also provided for the creation of alternative sub-structures in existing data sets. The default *part-of* relationships in the ontology data are used to link nodes in each graph; there are however other relationships that may be defined between components in each ontology, as [34, 14] describe. The third dimension allows *group* nodes to be drawn in planes parallel to a DAG, and used as the focus for the sub-set of nodes in a tree that together form a sub-structure with alternative relationships as defined by expert opinion and/or supporting literature.

Figures 6.29, 8.5, 8.6 and 8.7 provide illustrations of the visual analysis solutions developed to meet the user requirements described in § 5.3.

10.3 Conclusions

Three hypotheses were formulated during the course of this project, to guide the research being done and evaluation of the options developed for visual analysis. These were to examine the following:

1. any advantages visual analysis may provide over text-based analysis

2. whether visualisation in 3D provides advantages over 2D
3. the influence of spatial awareness or ability on visual analysis.

A number of factors influence the effectiveness of visual data analysis solutions. To be able to draw valid conclusions in the field within the limitations of this thesis, a decision was made to restrict the area of application. Analysis of anatomy ontology data in the EMAP and XSPAN projects provided a preliminary research area on which to focus, from which to extend results obtained to research in other biological ontologies and wider fields of application. A limited group of target users was available for evaluation of the work done, so that the hypotheses formulated cannot be rejected or accepted based on statistical significance. However, the research done and qualitative and quantitative information obtained during the heuristic and structured evaluations performed allow conclusions to be drawn that answer the research questions posed.

Extensive research was done into current methods available for analysis of especially hierarchically structured data, evaluating options for constructing overviews of data structure followed by detailed analysis of ROIs. Evidence in the literature points to the advantages obtained when visualisations are generated that harness highly developed perception in humans, mapping data structure to visual representations that aid users in building effective mental models of the information contained within data. Fairly strong evidence was also obtained from evaluations performed during the course of this project that supports significant advantage in visual analysis of the structured data this thesis examines. It is necessary to stress the importance of identifying and incorporating into visualisation solutions effective cues that are recognised and correctly interpreted by users who may have varying backgrounds, domain knowledge, and computing and other ability. Providing intuitive support for detailed analysis of ROIs contributes significantly to effective retrieval of knowledge contained within data and user satisfaction.

Each of 2D and 3D provides advantages for analysis over the other; this thesis has done a large amount of research on the advantages of visual analysis in different dimensions, including the partial dimensions between 2D and 3D. The main advantage in 2D is that it provides a simple interface that maps to the 2D *canvases* used in normal working life. Further, the current state of computing uses largely 2D for input and output; projection of visualisations created in higher dimensions onto 2D surfaces will be subject to some degree of distortion that relies on human perception to reconstruct the illusion of or projection back into higher dimensions correctly. However there are space limitations in 2D — research in information visualisation looks at techniques for overcoming these restrictions that quickly become apparent analysing large data sets, especially where a significant number of relationships exist within the data. Limitations in space in 2D present a problem for the approach used in this thesis to develop solutions to challenges identified in the analysis of the ontologies. For this case 3D clearly provides advantages that are not available in 2D, especially for simultaneous analysis of multiple data sets, justifying the extension of the 2D prototype to 3D. Evaluation results in chapters 7 and 9 indicate user preference for 3D, despite difficulty inherent in 3D

navigation, reported in literature in the field and experienced in the use of the prototypes developed. Further work is required to provide greater support for navigation in the 3D visualisation browser.

Research into the influence of human spatial ability on (perceived) usability of spatial analysis solutions and willingness and ability to explore information spaces deal mostly with gender and age differences. Although the focus of the evaluations performed was not on these two attributes the two factors were examined as part of the evaluation procedure. Neither of the two structured evaluations performed found either age or gender to have a significant influence on usability or user interaction with the 3D visualisations. It should be noted however that all users were aged between 20 and 40, while literature in the field normally evaluates spatial ability across a much wider age range. Gender was fairly evenly balanced in both evaluations. Domain knowledge was found to have the largest influence on users' search strategies and level of interaction with the visual structures.

Only simple tests of spatial ability were performed for the second evaluation, and fairly small variation was found between users for the exercises performed. Coupled with the small number of users not enough information was obtained to allow strong conclusions to be drawn on the influence of spatial ability or awareness on use of the visualisation browsers. Additional tests and more focused research are required to draw conclusions on the third research hypothesis.

10.4 Future work

10.4.1 Extending visual analysis solutions developed

Relationships between elements in ontologies generally describe a hierarchical structure, tending toward a DAG or a network as the number of relationships between data elements increases. The visualisations developed for this thesis use hierarchical node-link graphs to describe the structure of the anatomy ontologies studied, allowing users to trace paths between data elements that encode the relationships occurring between node pairs. The structures drawn may be fairly easily extended to analysis of ontologies other than those tested, by mapping the hierarchical structure of such data sets to the DAGs generated by the prototypes developed. Options for encoding properties of data nodes and the relationships defined between node pairs may be adapted or extended to aid identification of data attributes. Graphical support for displaying sub-structures describing additional relationships within data and the ability to draw links between elements across multiple data sets allow interactive editing of default visual structures where necessary, to provide more effective encoding of complex relationships in data.

A limitation of the current implementation is that it reads specific properties for each data node, requiring an element name and ID and additional, optional properties such as synonyms and abbreviations for data elements (§ 6.5.4 lists properties defined for data el-

ements in the test ontologies). The ability to store and display alternative properties that may be defined in other ontologies is necessary — the option to include such information is currently only possible as additional annotation of objects in each graph. More flexible methods for modifying attributes required to draw objects in the visualisations would simplify writing the additional data **Loaders** required to extend the options currently available for generating visualisations of ontologies (refer § 6.3 and figures 6.3 and 6.4).

Other options for further development to address specific problems encountered in use of the browsers have been discussed in § 9.5 and § 9.6. A final option to explore is testing alternatives for navigation in the 3D browser using on-screen controls and joy-sticks, to determine if either option or both used in concert would provide users with better control over navigation in the 3D window.

10.4.2 Potential solutions to performance limitations in Java

§ 9.6.1 summarises limitations in the approach used, including the performance problems associated with interactive Java applications. An alternative solution could (re)develop the visualisations browsers using OpenGL (with C/C++), to take advantage of advanced 3D modelling capabilities, hardware acceleration and the larger user base and support for OpenGL. *OpenGL for Java* bindings¹ (also known as *GL4Java* bindings) could then be used to provide an interface that takes advantage of the benefits of Java — cross-platform compatibility and dissemination on the web.

An alternative would be to rewrite the Java code to make use of the newly developed Java™ Bindings for OpenGL®, JOGL². This would enable calls to be made directly from Java to the native OpenGL libraries, to make use of more effective hardware support for rendering graphics while still making full use of the Swing libraries, and without the overhead associated with calls between C/C++ and Java.

10.4.3 Study of factors with subjective influence on visual analysis

Finally, further research into variation in human spatial ability/awareness and the influence it has on spatial analysis would provide more information on the development of cues that would increase intuitiveness of the analysis options provided. This would involve a wider range of standard tests for spatial ability and a much larger test group, to explore further human spatial ability and how this can be harnessed for intuitive information visualisation. The influence of domain knowledge on the use of the visual structures generated and differences in preference for 2D and 3D also present interesting directions in which to perform further research.

¹See OpenGL™ for Java™: <http://gl4java.sourceforge.net>

²See the JOGL Project at: <https://jogl.dev.java.net>

Appendix A

Sample input files

A.1 DTD for EMAP anatomy ontologies

```
<!ELEMENT HGU_MRC_Edinburgh (species, anatomy*)>
<!ELEMENT anatomy (stage, component*)>
<!ELEMENT component (parentId,lineageparent*,lineagechildren*,
    synonym*,abbreviation?,deletedflag?,printName?,component*)>

<!ELEMENT species (#PCDATA)>
<!ELEMENT printName (#PCDATA)>
<!ELEMENT lineageparent (#PCDATA)>
<!ELEMENT lineagechildren (#PCDATA)>
<!ELEMENT synonym (#PCDATA)>
<!ELEMENT abbreviation (#PCDATA)>
<!ELEMENT deletedflag (#PCDATA)>
<!ELEMENT parentId (#PCDATA)>

<!ATTLIST component
    name CDATA #REQUIRED
    id CDATA #REQUIRED>
<!ATTLIST stage
    name CDATA #REQUIRED>
```

A.2 DTD for user session XML files

```
<!ELEMENT AnatomyTreeLoaderFile (DevelopmentStage*, AbstractOrganism*,
mapping*)>
<!ELEMENT DevelopmentStage (StageType, Specie, EntryDate, RootID,
LevelsToDraw, GraphOrientation, LabelProperty, component*)>
<!ELEMENT AbstractOrganism (Specie, EntryDate, RootID,
LevelsToDraw, GraphOrientation, LabelProperty, component*)>
<!ELEMENT component (printName, abbreviation, synonym*, primaryParentID,
parentID+, childID*, startStage, stopStage, componentType, relationship*,
comment*, treeLevel, modes, component*)>
<!ELEMENT mapping (relationship+, comment*)>

<!ELEMENT StageType (#PCDATA)>
<!ELEMENT Specie (#PCDATA)>
<!ELEMENT EntryDate (#PCDATA)>
<!ELEMENT RootID (#PCDATA)>
<!ELEMENT LevelsToDraw (#PCDATA)>
<!ELEMENT GraphOrientation (#PCDATA)>
<!ELEMENT LabelProperty (#PCDATA)>

<!ELEMENT printName (#PCDATA)>
<!ELEMENT abbreviation (#PCDATA)*>
<!ELEMENT synonym (#PCDATA)>
<!ELEMENT primaryParentID (#PCDATA)>
<!ELEMENT parentID (#PCDATA)>
<!ELEMENT childID (#PCDATA)>
<!ELEMENT startStage (#PCDATA)>
<!ELEMENT stopStage (#PCDATA)>
<!ELEMENT componentType (#PCDATA)>
<!ELEMENT relationship (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT treeLevel (#PCDATA)>
<!ELEMENT modes EMPTY>
```

(cont'd on next page)

```
<!--ATTLIST DevelopmentStage name CDATA #REQUIRED>
<!--ATTLIST AbstractOrganism name CDATA #REQUIRED>
<!--ATTLIST component name CDATA #REQUIRED id    CDATA #REQUIRED>
<!--ATTLIST modes  m0    CDATA #REQUIRED
  m1    CDATA #REQUIRED
  m2    CDATA #REQUIRED
  m3    CDATA #REQUIRED
  m4    CDATA #REQUIRED
  m5    CDATA #REQUIRED
  m6    CDATA #REQUIRED
  m7    CDATA #REQUIRED
  m8    CDATA #REQUIRED
  m9    CDATA #REQUIRED>
<!--ATTLIST mapping
  componentName CDATA #REQUIRED
  componentID    CDATA #REQUIRED>
<!--ATTLIST relationship
  id    CDATA #IMPLIED
  componentName CDATA #IMPLIED
  componentID    CDATA #IMPLIED>
<!--ATTLIST comment
  date CDATA #REQUIRED
  time CDATA #REQUIRED>
```

A.3 Reloadable session file for TS11

This file includes a user-created group and user comments for two nodes, and was saved from the graph in figure 6.22.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnatomyTreeLoaderFile>
  <DevelopmentStage name="TS11">
    <StageType>Theiler Stage</StageType>
    <Specie>mouse</Specie>
    <EntryDate>7/8/2001</EntryDate>
    <RootID>0</RootID>
    <LevelsToDraw>7</LevelsToDraw>
    <GraphOrientation>10</GraphOrientation>
    <LabelProperty>20</LabelProperty>
    <component name="mesoderm" id="189">
      ...
    <component name="GroupNode" id="1234">
      <printName>embryo.ectoderm.embryo.ectoderm.neural ectoderm.GroupNode
</printName>
      <abbreviation/>
      <primaryParentID>151</primaryParentID>
      <parentID>151</parentID>
      <parentID>150</parentID>
      <parentID>170</parentID>
      <parentID>165</parentID>
      <childID>158</childID>
      <childID>156</childID>
      <childID>153</childID>
      <childID>173</childID>
      <startStage>11</startStage>
      <stopStage>13</stopStage>
      <componentType>32</componentType>
      <relationship id="165">'part-of'</relationship>
      <relationship id="151">'part-of'</relationship>
      <relationship id="150">'part-of'</relationship>
      <relationship id="170">'part-of'</relationship>
      <treeLevel>4</treeLevel>
      <modes m9="0" m8="0" m7="0" m6="0" m5="0" m4="1" ...m1="0" m0="0"/>
    </component>
  ...
</AnatomyTreeLoaderFile>
```

```

...
    <component name="ectoderm" id="188">
        <printName>extraembryonic component.chorion.ectoderm</printName>
        <abbreviation/>
        <primaryParentID>187</primaryParentID>
        <parentID>187</parentID>
        <startStage>11</startStage>
        <stopStage>12</stopStage>
        <componentType>30</componentType>
        <relationship id="187">'part-of'</relationship>
        <comment date="18-Feb-2006" time="15h52m22">ASD: annotated node</comment>
        <treeLevel>2</treeLevel>
        <modes m9="0" m8="0" m7="0" m6="0" m5="0" m4="1" ...m1="0" m0="0"/>
    </component>
...
    <component name="endoderm" id="190">
        <printName>extraembryonic component.endoderm</printName>
        <abbreviation/>
        <primaryParentID>176</primaryParentID>
        <parentID>176</parentID>
        <childID>191</childID>
...
        <componentType>30</componentType>
        <relationship id="176">'part-of'</relationship>
        <treeLevel>1</treeLevel>
        <modes m9="0" m8="0" m7="0" m6="0" m5="0" m4="1" ...m1="0" m0="0"/>
    </component>
    <component name="Root - TS11" id="0">
        <printName>Root - TS11</printName>
        <abbreviation/>
        <primaryParentID>-1</primaryParentID>
        <parentID>-1</parentID>
        <childID>147</childID>
        <childID>176</childID>
        <startStage>-1</startStage>
        <stopStage>-1</stopStage>
        <componentType>30</componentType>
        <treeLevel>-1</treeLevel>
        <modes m9="0" m8="0" m7="0" m6="0" m5="0" m4="0" ...m1="0" m0="0"/>
    </component>
</DevelopmentStage>
</AnatomyTreeLoaderFile>

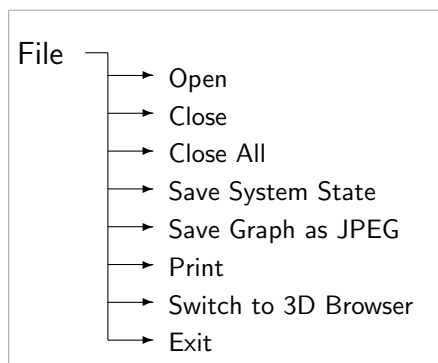
```

Appendix B

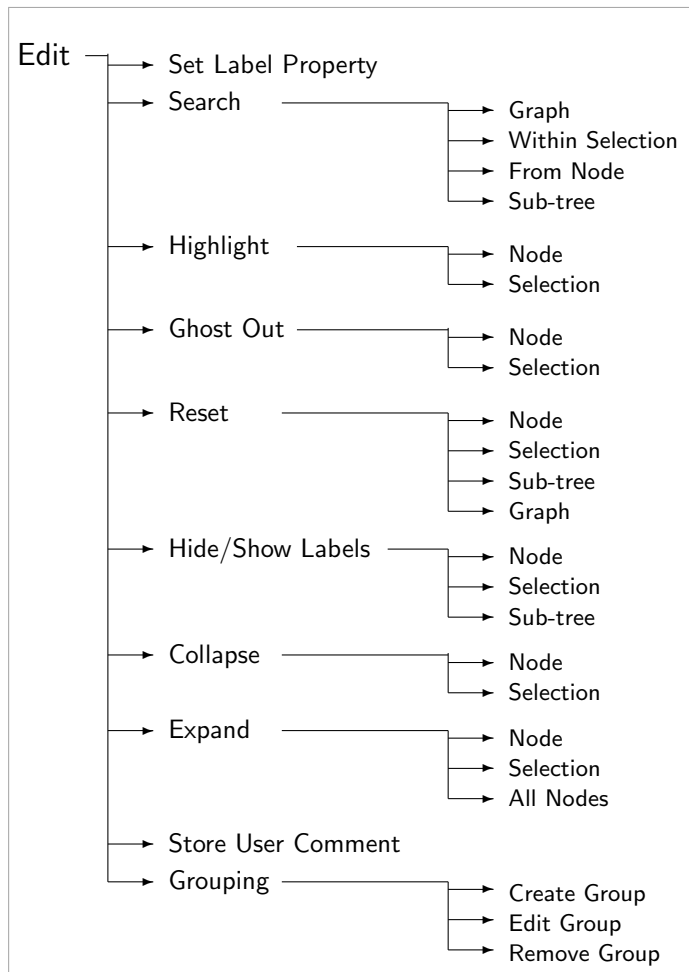
Design documents

B.1 Application menu for 2D browser

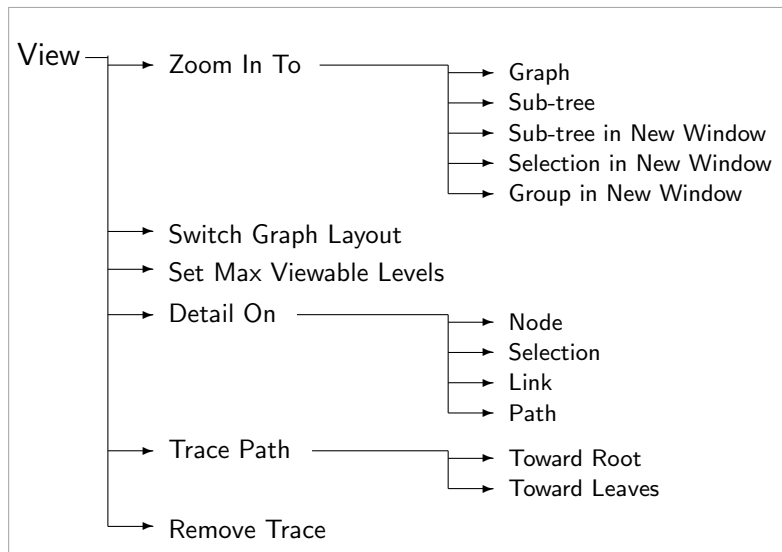
File menu



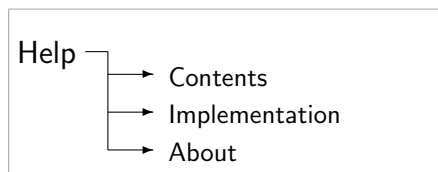
Edit menu



View menu

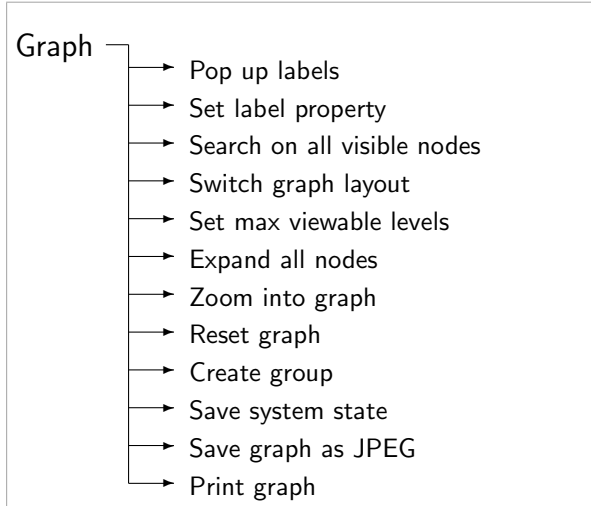


Help menu

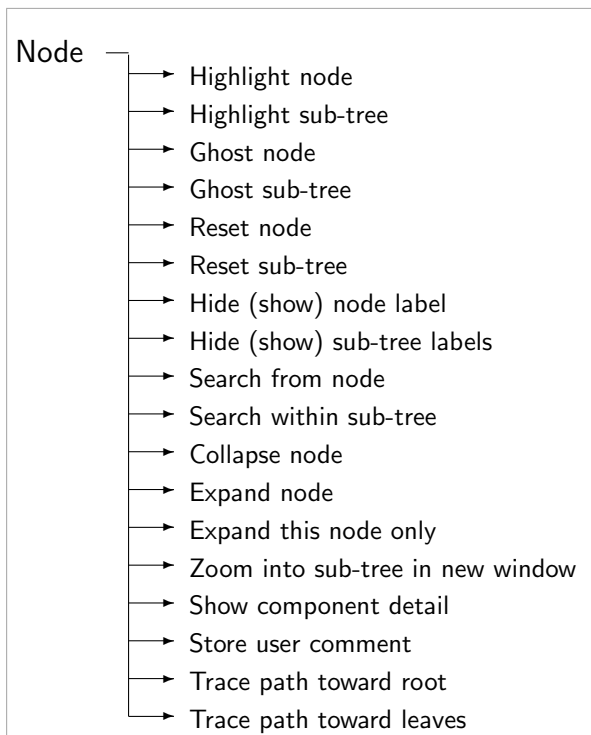


B.2 Popup menus for 2D browser

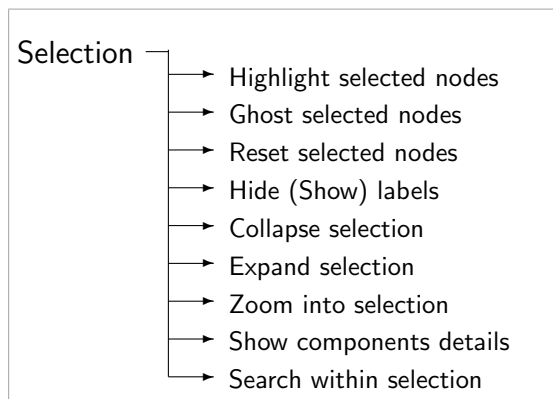
Graph popup menu



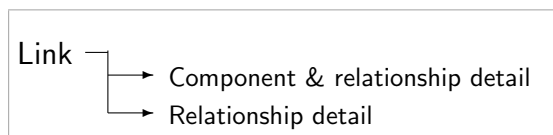
Node popup menu



Selection popup menu

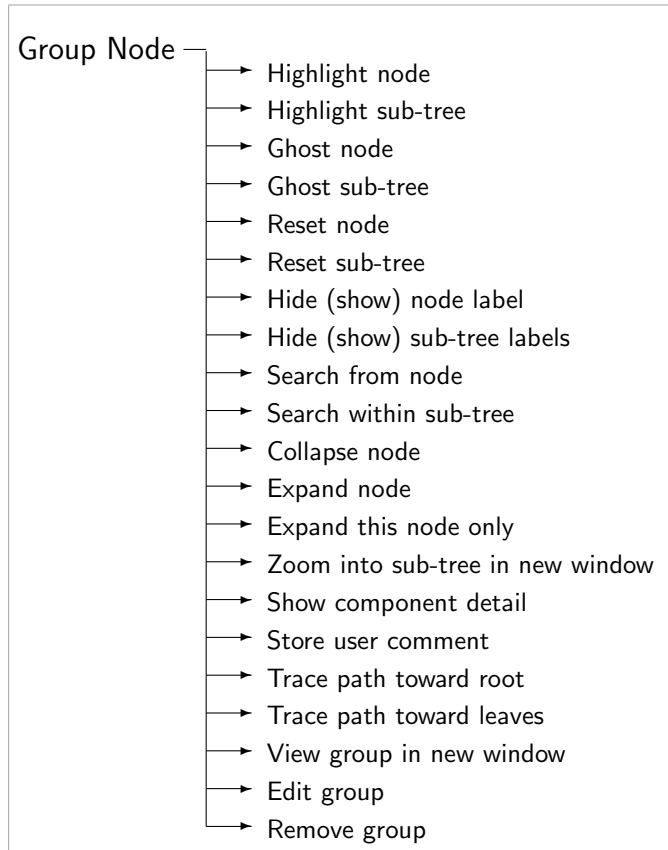


Link popup menu



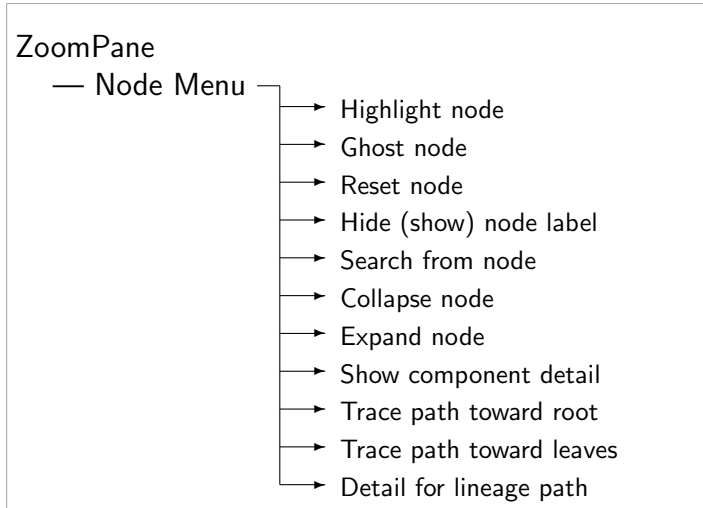
Group node popup menu

Note that the popup menu brought up for group node is more restricted than that for a node.

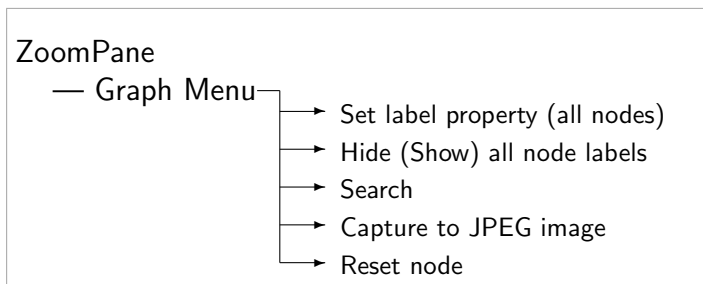


B.3 Zoom pane popup menus

Node menu

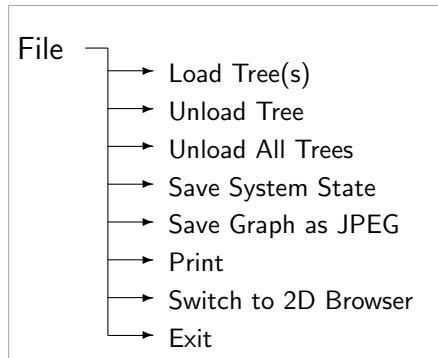


Graph menu

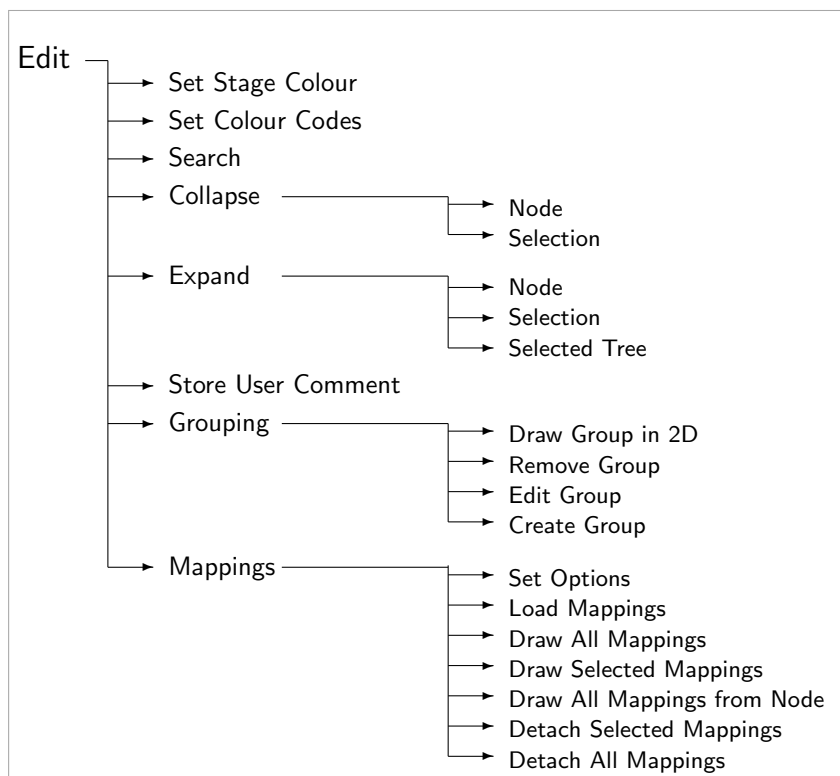


B.4 Application menu for 3D browser

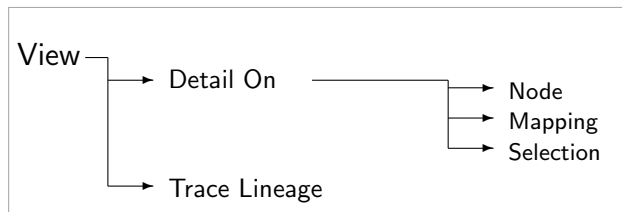
File menu



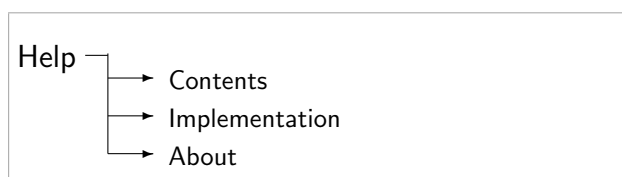
Edit menu



View menu

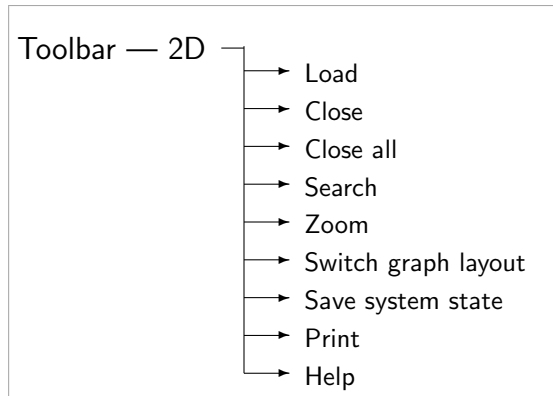


Help menu

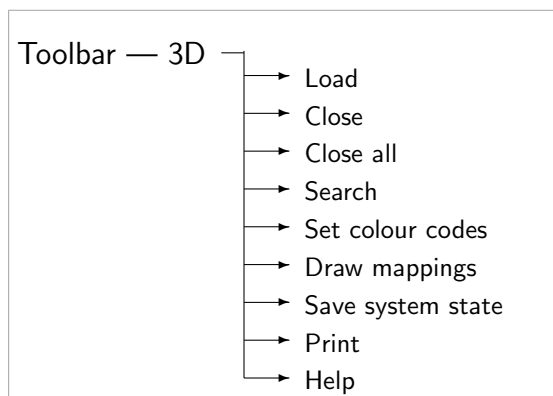


B.5 Toolbars for 2D and 3D browsers

Toolbar for 2D browser



Toolbar for 3D browser



B.6 Navigation aids for the 3D browser

B.6.1 Actions associated with *MouseBehaviors*

Mouse action	System response
hold and drag left mouse button	<i>rotation</i> round central axis
hold and drag right mouse button	<i>translation</i> along plane in which mouse is moved
hold and drag middle mouse button	<i>zoom</i> in and out of 3D scene

B.6.2 Actions associated with *KeyNavigatorBehaviors*

Keyboard action	System response	System response with Alt-Key depressed
left/right arrow key	<i>rotation</i> round viewpoint central axis in direction of arrow (note that actual objects in scene will move in opposite direction)	<i>translation</i> along plane in direction of arrow
page up/down	<i>rotation</i> round viewpoint central axis upwards or downwards respectively	<i>translation</i> upwards or downwards respectively
up/down arrow key	<i>zoom</i> in and out of 3D scene respectively	N/A
'=' key	return to default viewpoint at centre of universe	N/A

Appendix C

Evaluation documents

C.1 User instruction sheet

Instructions for User Evaluation	User ID:
<p>Thank you for agreeing to participate in the evaluation of the two anatomy browsers. All data obtained in the evaluation is confidential. Although it would be helpful to allow us to get back to you with any additional questions we may have, you are welcome to omit your name and contact details if you would prefer the data to remain anonymous. You can withdraw from the evaluation and request that your data be destroyed at any stage. All data storage will comply with the appropriate Data Protection regulations. At the end of the evaluation process we will provide all users with feedback on the results unless requested otherwise.</p> <p>Please fill in the user questionnaire provided. This provides us with extremely useful background information on users, their work and the technologies they use.</p> <p>Once you have finished filling out the questionnaire you will be provided with Task Scenario Sheets. Please use these with the help of the printout of the Quick Guide to complete the tasks detailed. Where required write out responses to questions asked on the Task Scenario sheets. More detailed help is provided from the Help Menu in each browser, and you may ask the evaluator for clarifications where necessary, and as much assistance as necessary will be given provided doing so does not bias the results of the evaluation. The evaluator will make notes on the path(s) you take to your solution for each task. A talk-through of the process you follow to achieve each goal would be appreciated as it provides more information on your understanding of how the systems work.</p> <p>Once you have completed the tasks you will be provided with two further questionnaires to gather your impressions on the functionality of the system. You are welcome at this stage to provide any further information on the system that you feel has not been addressed sufficiently in the questionnaires.</p> <p>If you agree to us contacting you for any additional information, please tick here: <input type="checkbox"/></p> <p>If you would like to receive feedback at the end of the analysis, please tick here: <input type="checkbox"/></p> <p>If you understand and accept the above, please sign below.</p> <p>Signature:</p> <p>Name:</p> <p>Date:</p>	

C.2 Task scenario sheets

Key

REQ	Information / equipment required to carry out task
SCC	Successful completion criteria
MTC	Maximum time to complete task

2D Browser

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
1	Load Theiler Stage (TS) 11 in the browser	REQ: 2D anatomy browser, Quick Guide SCC: Visualisation of TS11 displayed in browser MTC: 10s	N/A
2	Identify the anatomy component chorion and list the components which are 'part-of' chorion (immediate children of), as well as the Theiler Stages through which they persist.	REQ: 2D anatomy browser, Quick Guide SCC: Expansion of the DAG to show at least the component chorion. Components that are 'part-of' the chorion may be identified by tracing down the tree. An alternative is to bring up the component detail for chorion and list its children. MTC: 60s	ID: 188 ectoderm Stgs 11-12 ID: 189 mesoderm Stgs 11-11
3	Determine if any of the components identified in step 2 above have synonyms used to refer to them, and if so, what they are.	REQ: 2D anatomy browser, Quick Guide SCC: List of synonyms for each component obtained by bringing up component detail for the components. An alternative is to highlight the components required and switch label property to "component synonym". MTC: 15s	None

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
4	Identify the component amnion (print name - extraembryonic component.amnion) and display its complete sub-tree, while suppressing all others. Determine the depth of its sub-tree and identify the stages through which its leaf nodes persist.	REQ: 2D anatomy browser, Quick Guide SCC: Display of sub-tree in a separate window, or in isolation in the main window. Component detail of leaf nodes to retrieve Embryo Start and Stop Stages. MTC: 30s	Depth - 1 Leaf nodes: ID: 181 mesoderm Stgs 11-11 ID: 180 ectoderm Stgs 11-12
5	Load TS12 in the browser. Using the top-down or left-right layout of the DAG highlight the component branchial arch and determine how many components make up the branchial arch (immediate children of).	REQ: 2D anatomy browser, Quick Guide SCC: Visualisation of TS12 displayed in browser using either the top-down or left-right layout. Correct number of components that are 'part-of' the branchial arch identified. (This may require search, expansion of sub-tree and/or bringing up component detail.) MTC: 55s	1 component - (1st arch)
6	Switch the layout to radial and highlight the component gut. Identify the number of components derived from the gut (immediate children of), and list their print names and the Theiler Stages through which they persist.	REQ: 2D anatomy browser, Quick Guide SCC: Visualisation of TS12 displayed in browser using the radial layout. Correct number of components that are 'part-of' the gut, their print names and the Theiler Stages through which they persist. (This may require search, expansion of sub-tree and/or bringing up component detail.) MTC: 250s	3 components: ID: 360 embryo.organ system.visceral or- gan.alimentary sys- tem.gut.hindgut diverticulum Stgs: 12 - 20 ID: 357 embryo.organ system.visceral organ.alimentary system.gut.foregut diverticulum Stgs: 12 - 13 ID: 364 embryo.organ system.visceral organ.alimentary system.gut.midgut Stgs: 12 - 23

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
7	Highlight all components that contain the term ectoderm as part of their print name.	REQ: 2D anatomy browser, Quick Guide SCC: Visualisation of TS12 displayed in browser. All components satisfying query highlighted using any method (easiest option uses search dialog). MTC: 20s	N/A
8	<p>Create a group node with the following properties: Name: GroupNode Print Name: PrintName ID: 1234 Embryo Start Stage: 9 Embryo Stop Stage: 13 Parents: ID: 391 extraembryonic component.trophectoderm.polar trophectoderm ID: 394 extraembryonic component.trophectoderm ID: 384 extraembryonic component.chorion ID: 270 embryo.mesenchyme ID: 390 extraembryonic component.endoderm.visceral endoderm Children IDs: ID: 395 extraembryonic component.trophectoderm.polar trophectoderm.ectoplacental cone ID: 286 embryo.mesenchyme.-trunk mesenchyme.mesenchyme derived from neural crest ID: 288 embryo.mesenchyme.-trunk mesenchyme.paraxial mesenchyme.somite ID: 276 embryo.mesenchyme.-head mesenchyme.paraxial mesenchyme.somite</p> <p>Hide nodes or collapse subtrees, and/or ghost out nodes as necessary to minimise crossing of links and occlusion of nodes. Save a copy of the canvas to file that shows the grouping of the data.</p>	REQ: 2D anatomy browser, Quick Guide SCC: Group drawn on DAG, JPEG file capturing contents of canvas MTC: 330s	N/A

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
9	Close all windows currently open and load TS26. Switch to radial view and display all nodes in the DAG. Identify the component respiratory tract and trace its ancestors toward the root and list all their component names.	REQ: 2D anatomy browser, Quick Guide SCC: Display of TS26 in radial view. Highlighting of respiratory tract and physical tracing of its lineage and correct identification of component names of ancestors. MTC: 300s	from the root: embryo organ system visceral organ respiratory system
10	Switch to the top-down layout and determine the number of components that contain the term dorsal in their component name.	REQ: 2D anatomy browser, Quick Guide SCC: Display of DAG using top-down layout. Identification of correct number of components satisfying query - search dialog gives count and list. MTC: 300s	number of components: 17

3D Browser

	TASK DESCRIPTION	TASK DETAIL
1	Load TS02, TS05, TS11 and TS14 in the browser.	REQ: 3D anatomy browser, Quick Guide SCC: Display of the 4 Stages in the browser MTC: 20s
2	Remove TS05 from the window and load TS08.	REQ: 3D anatomy browser, Quick Guide SCC: Removal of the DAG representing TS05 and loading of TS08 MTC: 15s
3	Zoom into TS11 and create a group node with the following properties: Name: GroupNode Print Name: PrintName ID: 1234 Embryo Start Stage: 9 Embryo Stop Stage: 13 Parent IDs: (ID: 168) embryo.notochordal plate ID: 167 embryo.mesoderm ID: 151 embryo.ectoderm.neural ectoderm ID: 150 embryo.ectoderm Children IDs: (ID: 173) embryo.organ system.cardiovascular system.heart.cardiogenic plate ID: 172 embryo.organ system.cardiovascular system.heart ID: 155 embryo.ectoderm.neural ectoderm.-future spinal cord ID: 153 embryo.ectoderm.neural ectoderm.-future brain.neural fold ID: 152 embryo.ectoderm.neural ectoderm.-future brain	REQ: 3D anatomy browser, Quick Guide SCC: Display of the group MTC: 260s
4	Determine the number of components containing the term extraembryonic in the component name	REQ: 3D anatomy browser, Quick Guide SCC: Correct number of components identified (4). Easiest solution is to use the search dialog MTC: 27s
5	Create a successive series of links between nodes (and across stages): TS02: (ID: 6) two-cell stage -> TS08: (ID: 73) extraembryonic component.-trophectoderm -> TS11: (ID: 150) embryo.ectoderm -> TS14: (ID: 702) embryo.limb.forelimb bud	REQ: 3D anatomy browser, Quick Guide SCC: Display of cross-stage links MTC: 169s

C.3 Questionnaires

C.3.1 Pre-evaluation questionnaire

Survey of Users (Please tick the appropriate options) User ID:

Name: Date:
 Email address:

1. Please indicate your age:

<20 years ☐ 20-29 years ☐ 30-39 years ☐ > 39 years ☐

2. Which are the best descriptions of your qualifications: (if more than one, indicate all)

	Undergraduate	MSc	PhD
Computing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Biology	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Genetics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bioinformatics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other (please specify)

3. Which is the best description of your current area of work or research?

Computing/IT ☐ Biology ☐ Bioinformatics ☐ Other ☐ Please specify:

4. Please indicate any previous research or work areas:

Computing/IT ☐ Biology ☐ Bioinformatics ☐ Other ☐ Please specify:

5. Please indicate what type of computer(s) you use at work:

(if more than one, indicate all)

Hardware	IBM Compatible PC <input type="checkbox"/>	Macintosh <input type="checkbox"/>	UNIX Box <input type="checkbox"/>	Other/ Don't know <input type="checkbox"/>
Processor Speed
Hard Disc size
Operating System	Windows 2000/ME <input type="checkbox"/>	Mac OS X <input type="checkbox"/>	Unix <input type="checkbox"/>	Please specify:
	Windows NT <input type="checkbox"/>	Mac OS 9.x <input type="checkbox"/>	Linux <input type="checkbox"/>
	Windows 98 <input type="checkbox"/>	Mac OS 8.x <input type="checkbox"/>	<input type="checkbox"/>
	Windows 95 <input type="checkbox"/>	Mac OS 7.x <input type="checkbox"/>	<input type="checkbox"/>	
	Windows 3.x <input type="checkbox"/>			
	Don't know <input type="checkbox"/>			

6. Please indicate what web browser and version you use for work:
(if more than one, indicate all)
- | | | | | | |
|----------------|--------------------------------------------|-----------------------------------|----------------------------------|---------------------------------|-------------------|
| Browser | Internet Explorer <input type="checkbox"/> | Netscape <input type="checkbox"/> | Mozilla <input type="checkbox"/> | Other <input type="checkbox"/> | (Please specify:) |
| | Latest <input type="checkbox"/> | Latest <input type="checkbox"/> | Latest <input type="checkbox"/> | Latest <input type="checkbox"/> | |
| please specify | Older <input type="checkbox"/> | Older <input type="checkbox"/> | Older <input type="checkbox"/> | Older <input type="checkbox"/> | |
7. Please indicate what type of connection you use for work:
(if more than one, indicate most used)
- | | | | | | |
|----------------|-----------------------------------|-------------------------------------|-------------------------------------|--------------------------------------|-------------------------------------|
| Type | LAN <input type="checkbox"/> | ADSL <input type="checkbox"/> | ISDN <input type="checkbox"/> | Phone modem <input type="checkbox"/> | Don't know <input type="checkbox"/> |
| Network | Academic <input type="checkbox"/> | NHS <input type="checkbox"/> | Commercial <input type="checkbox"/> | Other <input type="checkbox"/> | Don't know <input type="checkbox"/> |
| Speed | 1Gb <input type="checkbox"/> | 100 Base T <input type="checkbox"/> | 10 Base T <input type="checkbox"/> | Other <input type="checkbox"/> | Don't know <input type="checkbox"/> |
8. Please indicate the level of your skill in the use of computers:
- | | | | | | | |
|--------------------------------------------|--------------------------|------------|---|---|---|-------------|
| General computing | None | Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |
| Skill in use of data analysis tools | None | Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |
| Skill in use of visualisation tools | None | Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |
9. Please indicate which of the options below you use computers for:
(if more than one, indicate all)
- | | | | | | | |
|--------------------------|--------------------------|--------------------------|---------------------------|--------------------------|--------------------------|----------------|
| Word processing | Data Processing | Data Analysis | Data visualisation | Program-ming | Other | Please specify |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
10. Please indicate the types of data analysis tools you use, if any:
(if more than one, indicate all)
- | | | | |
|--------------------------|--------------------------|--------------------------|----------------|
| Graphical | Textual | Other | Please specify |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
11. Please indicate which of the options below apply best to your use of data analysis tools:
- | | | | | | | |
|------------------------------------------------|--------------------------|-----------------|---|---|---|-------------|
| Complexity of data analysis performed | None | Very Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |
| Frequency of use of data analysis tools | No use | Very Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |
| Usefulness of data analysis tools | Not useful | Very Low | | | | High |
| | <input type="checkbox"/> | 1 | 2 | 3 | 4 | 5 |

12. Please list the 3 data analysis tools you use most often, in decreasing order of preference:

1. _____
2. _____
3. _____

13. Please indicate whether or not you make use of data visualisation tools for your work:

Yes	No
<input type="checkbox"/>	<input type="checkbox"/>

14. Please indicate, if applicable, which of the options below apply best to your use of visualisation tools:

Complexity of visualisations generated	None	Very Low					High
	<input type="checkbox"/>	1	2	3	4	5	
Frequency of use of data visualisation tools	No use	Very Low					High
	<input type="checkbox"/>	1	2	3	4	5	
Usefulness of data visualisation tools	Not useful	Very Low					High
	<input type="checkbox"/>	1	2	3	4	5	

15. Please list the 3 data visualisation tools you use most often, in decreasing order of preference:

1. _____
2. _____
3. _____

16. How frequently do you use the currently working EMAP (Mouse Atlas) browsers?

Never	Occasionally	Monthly	Weekly	Daily
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

17. For how long have you been using the EMAP (Mouse Atlas) browsers?

< 1 mth	1-3 mths	3-6 mths	6 mths-1yr	> 1 yr	N/A
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

18. Please list in decreasing order of usefulness, your 5 most preferred features of the EMAP (Mouse Atlas) browsers:

1. _____
2. _____
3. _____
4. _____
5. _____

19. Please list in decreasing order of usefulness, up to 5 features you wish to see added to the current EMAP (Mouse Atlas) browsers, or extended:

1. _____
2. _____
3. _____
4. _____
5. _____

20. Please add any further comments or information that may be helpful:

Thank you for your help in completing the questionnaire.
I can be contacted at: ceedad@macs.hw.ac.uk

C.3.2 Post-evaluation questionnaire

Usability Evaluation Questionnaire^a

Identification number:

Age:

Sex: ☐ Female ☐ Male

Part 1: Type of System being Rated

1.1 Did you feel you had sufficient time to familiarise yourself with the new system?

Yes ☐

No ☐

1.2 On average, how much time would you spend per week on this system?

Less than 1 hour ☐

1 to less than 4 hours ☐

4 to less than 10 hours ☐

Over 10 hours ☐

Part 2: Past Experience

2.1 Of the following devices, software, and systems, check those that you have personally used and are familiar with.

Keyboard ☐

Electronic mail ☐

Numeric key pad ☐

Graphics software ☐

Mouse ☐

Computer games ☐

Light pen ☐

Colour monitor ☐

Touch screen ☐

Time-share system ☐

Track ball ☐

Workstation ☐

Joy stick ☐

Personal computer ☐

Text editor ☐

Floppy drive ☐

Word processor ☐

Hard drive ☐

File manager ☐

Compact disk drive ☐

Electronic spreadsheet ☐

cont'd on next page

^aCopyright©1988, 1989, 1991 Human-Computer Interaction Laboratory, University of Maryland. All rights reserved. The original (Shneiderman) Usability Evaluation questionnaire has been adapted to suit this evaluation.

Part 3: Overall reactions to the system

3.1	Terrible									Wonderful	
	1	2	3	4	5	6	7	8		9	N/A
3.2	Frustrating									Satisfying	
	1	2	3	4	5	6	7	8		9	N/A
3.3	Dull									Stimulating	
	1	2	3	4	5	6	7	8		9	N/A
3.4	Difficult									Easy	
	1	2	3	4	5	6	7	8		9	N/A
3.5	Inadequate power									Adequate power	
	1	2	3	4	5	6	7	8		9	N/A
3.6	Rigid									Flexible	
	1	2	3	4	5	6	7	8		9	N/A

cont'd on next page

Part 4: Data Visualisation & Screen

4.1 How representative are the visualisations generated of your mental model of the data structure?

Not at all

Very much so

1 2 3 4 5 6 7 8 9 N/A

4.2 Do you find that the visualisations of the data provide an advantage over the textual indices in current use?

Not at all

Very much

1 2 3 4 5 6 7 8 9 N/A

For each of the options 4.3-4.7, compare the ease of use of the visualisations generated to the textual indices in current use:

4.3 Data structure

**Visualisations more
difficult to use**

**Visualisations easier
to use**

1 2 3 4 5 6 7 8 9 N/A

4.4 Understanding of data

**Visualisations more
difficult to use**

**Visualisations easier
to use**

1 2 3 4 5 6 7 8 9 N/A

4.5 Search and query

**Visualisations more
difficult to use**

**Visualisations easier
to use**

1 2 3 4 5 6 7 8 9 N/A

4.6 Tracing lineage

**Visualisations more
difficult to use**

**Visualisations easier
to use**

1 2 3 4 5 6 7 8 9 N/A

4.7 Grouping of data

**Visualisations more
difficult to use**

**Visualisations easier
to use**

1 2 3 4 5 6 7 8 9 N/A

4.8 How easy was it to read text on the computer screen?

Hard to read

Easy to read

1 2 3 4 5 6 7 8 9 N/A

4.9 How easy was it to navigate through the data structure?

Difficult

Intuitive

1 2 3 4 5 6 7 8 9 N/A

4.10 How would you rate occlusion of data in the visualisations generated?

High clutter

Low clutter

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

Please rate the use of the Left-Right layout of the data									
4.11	Difficult to use							Intuitive	
	1	2	3	4	5	6	7	8	9
									N/A
4.12	High clutter							Low clutter	
	1	2	3	4	5	6	7	8	9
									N/A
Please rate the use of the Top-Down layout of the data									
4.13	Difficult to use							Intuitive	
	1	2	3	4	5	6	7	8	9
									N/A
4.14	High clutter							Low clutter	
	1	2	3	4	5	6	7	8	9
									N/A
Please rate the use of the Radial layout of the data									
4.15	Difficult to use							Intuitive	
	1	2	3	4	5	6	7	8	9
									N/A
4.16	High clutter							Low clutter	
	1	2	3	4	5	6	7	8	9
									N/A
Please show the level of usefulness of each of the options available (4.17-4.21) for the reduction of occlusion									
4.17 Hiding of labels									
	Unuseful							Useful	
	1	2	3	4	5	6	7	8	9
									N/A
4.18 Ghosting of nodes									
	Unuseful							Useful	
	1	2	3	4	5	6	7	8	9
									N/A
4.19 Hiding of sub-trees									
	Unuseful							Useful	
	1	2	3	4	5	6	7	8	9
									N/A
4.20 Zoom									
	Unuseful							Useful	
	1	2	3	4	5	6	7	8	9
									N/A
4.21 Switching between layouts									
	Unuseful							Useful	
	1	2	3	4	5	6	7	8	9
									N/A
4.22 Location of information required									
	Difficult to find							Easy to find	
	1	2	3	4	5	6	7	8	9
									N/A

cont'd on next page

4.23 Performance of search and query operations									
Difficult								Easy	
1	2	3	4	5	6	7	8	9	N/A
4.24 Interpretation of search and query results									
Difficult								Easy	
1	2	3	4	5	6	7	8	9	N/A
4.25 Does the visualisation of the data ease determination of lineage, compared to the system currently in place for determining lineage?									
No difference								More intuitive	
1	2	3	4	5	6	7	8	9	N/A
4.26 How well does the method provided for grouping of data highlight user-specified data groups?									
Poorly highlighted								Well highlighted	
1	2	3	4	5	6	7	8	9	N/A

cont'd on next page

Part 5: Terminology and System Information

5.1 Use of terms throughout system

Inconsistent

Consistent

1 2 3 4 5 6 7 8 9 N/A

5.2 Does the terminology relate well to the work you are doing?

Unrelated

Well related

1 2 3 4 5 6 7 8 9 N/A

5.3 Messages which appear on screen

Inconsistent

Consistent

1 2 3 4 5 6 7 8 9 N/A

5.4 Messages which appear on screen

Confusing

Clear

1 2 3 4 5 6 7 8 9 N/A

5.5 Does the computer keep you informed about what it is doing?

Never

Always

1 2 3 4 5 6 7 8 9 N/A

5.6 Error messages

Unhelpful

Helpful

1 2 3 4 5 6 7 8 9 N/A

5.7 System feedback

Unhelpful

Helpful

1 2 3 4 5 6 7 8 9 N/A

5.8 Ability to identify errors and sources of errors

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

5.9 System help/support

Not useful

Useful

1 2 3 4 5 6 7 8 9 N/A

5.10 Level of system support for error recovery

Low

High

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

Part 6: Learning

6.1 Understanding of terms used throughout system

Difficult

1 2 3 4 5 6 7 8

Easy

9 N/A

6.2 Does the terminology relate well to the work you are doing?

Discouraging

1 2 3 4 5 6 7 8

Encouraging

9 N/A

6.3 Understanding of messages which appear on screen

Difficult

1 2 3 4 5 6 7 8

Easy

9 N/A

6.4 Usefulness of messages which appear on screen

Never

1 2 3 4 5 6 7 8

Always

9 N/A

6.5 Error messages

Confusing

1 2 3 4 5 6 7 8

Clear

9 N/A

6.6 Does the computer keep you informed about what it is doing?

Confusing

1 2 3 4 5 6 7 8

Clear

9 N/A

6.7 Ease of learning of the functions available

Difficult

1 2 3 4 5 6 7 8

Easy

9 N/A

6.8 Actual ability to make use of the system

Difficult

1 2 3 4 5 6 7 8

Easy

9 N/A

6.9 How would you rate the time you required, on average, to perform tasks?

Very long

1 2 3 4 5 6 7 8

Very short

9 N/A

6.10 How long do you feel you would require to reach a working level of proficiency?

> 1 year

1 year

6 mths

1 mth

2 weeks

1 day

cont'd on next page

Part 7: System Capabilities

7.1 System speed, on average

Too slow

Fast enough

1 2 3 4 5 6 7 8 9 N/A

7.2 Variations in system speed

Large

Small

1 2 3 4 5 6 7 8 9 N/A

7.3 How reliable is the system?

Very unreliable

Very reliable

1 2 3 4 5 6 7 8 9 N/A

7.4 System tends to be

Noisy

Quiet

1 2 3 4 5 6 7 8 9 N/A

7.5 Correcting your mistakes

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

7.6 Are the needs of both experienced and inexperienced users taken into account?

Never

Always

1 2 3 4 5 6 7 8 9 N/A

7.7 How would you rate the level of functionality offered by the system?

Poor

Very good

1 2 3 4 5 6 7 8 9 N/A

Compared to the current working browsers how would you rate this system:

7.8

Difficult to use

Easy to use

1 2 3 4 5 6 7 8 9 N/A

7.9

**Difficult data
analysis**

**Simplified data
analysis**

1 2 3 4 5 6 7 8 9 N/A

7.10

Unintuitive

Intuitive

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

Part 8: Users' Comments

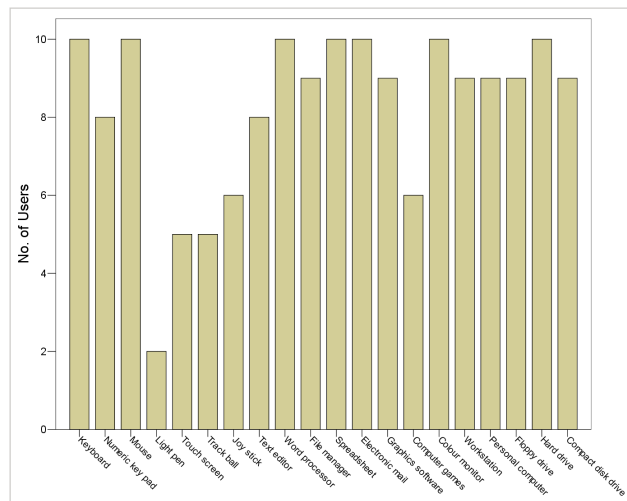
Please write any comments you have in the space below.

Appendix D

Evaluation results

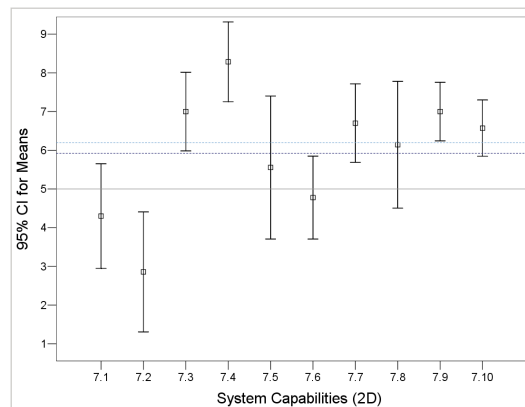
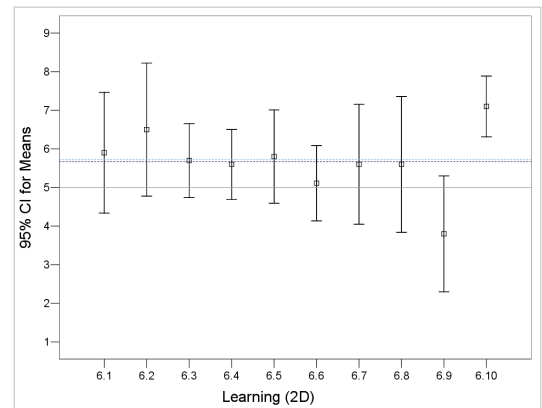
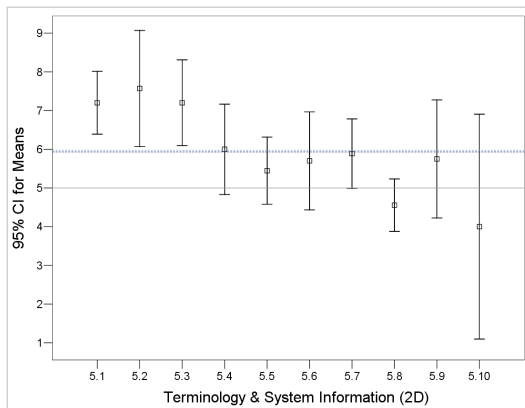
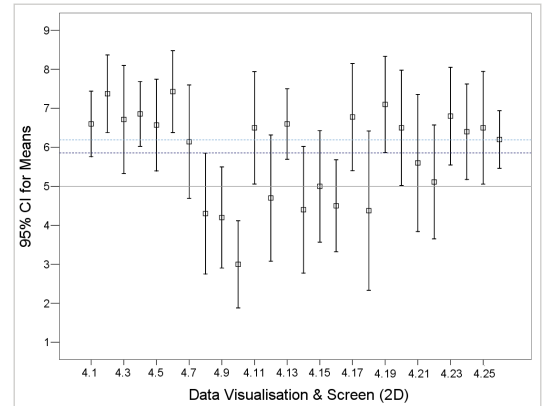
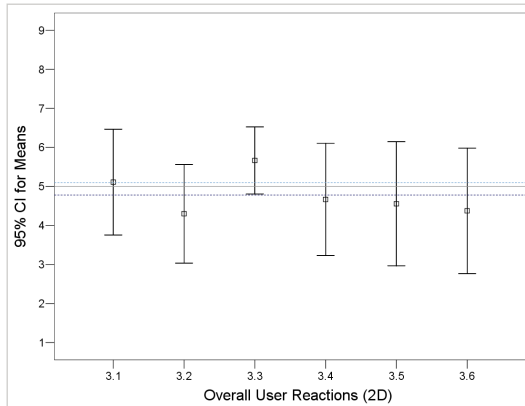
D.1 Pre-evaluation questionnaire

D.1.1 Use of input/output devices

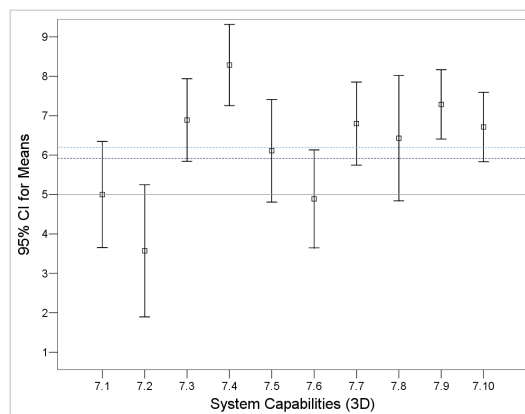
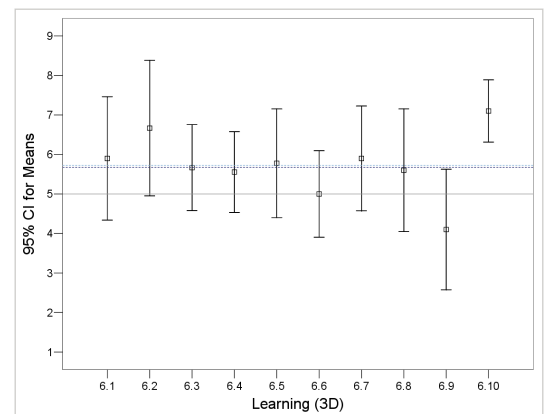
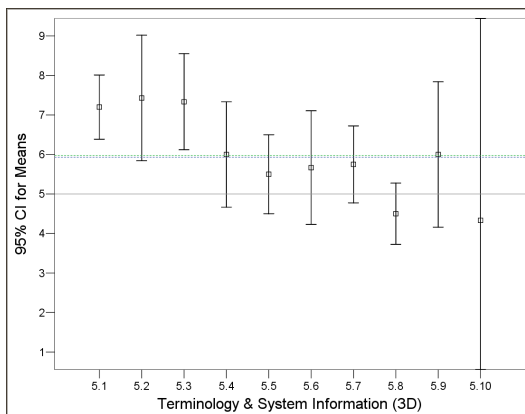
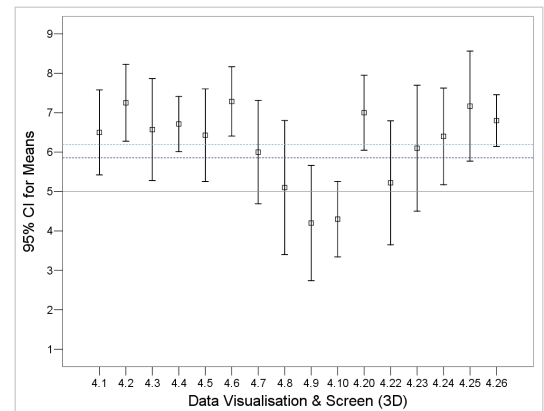
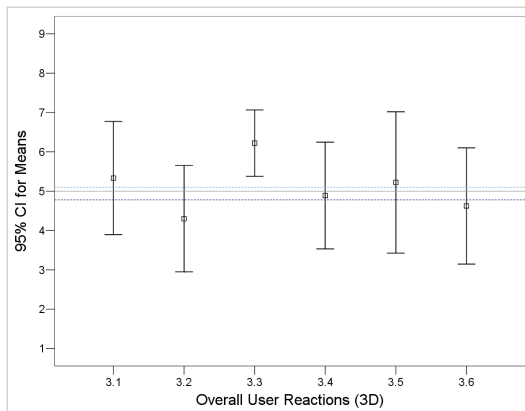


D.2 Post-evaluation questionnaire

D.2.1 Mean rankings over all users for each item for 2D browser

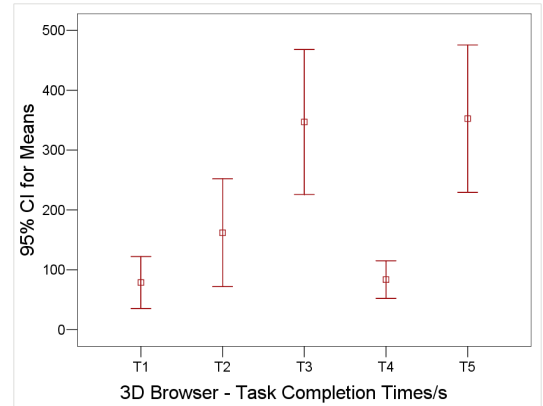
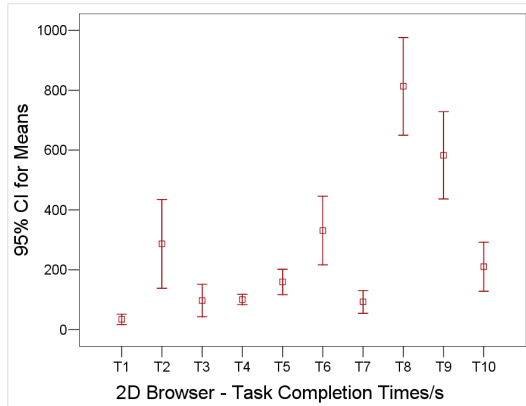


D.2.2 Mean rankings over all users for each item for 3D browser

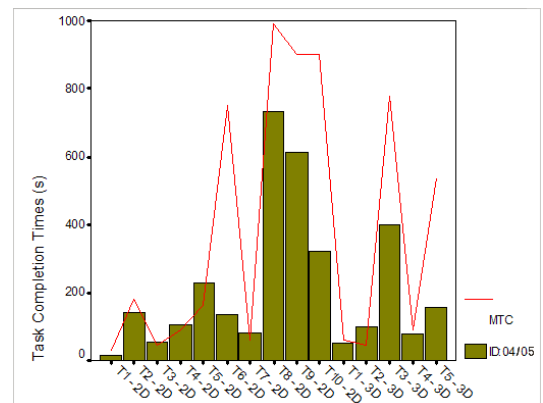
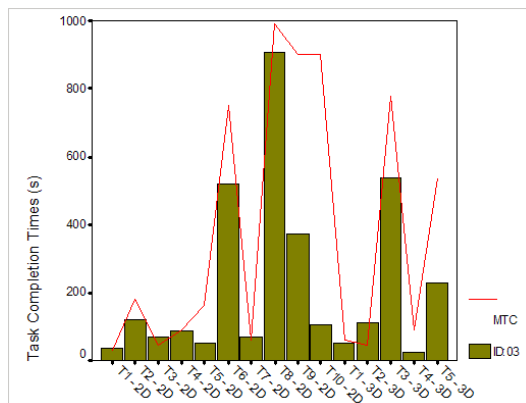
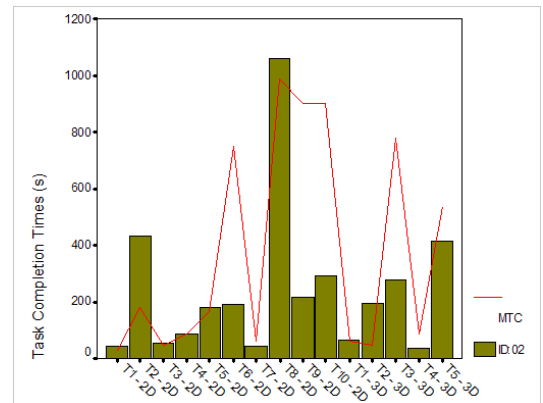
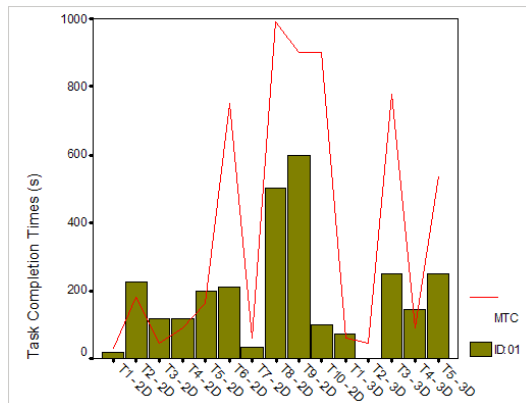


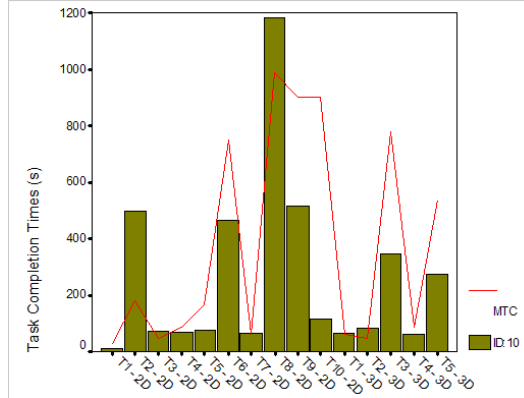
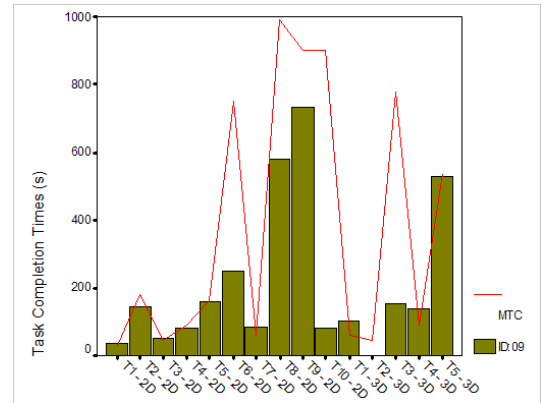
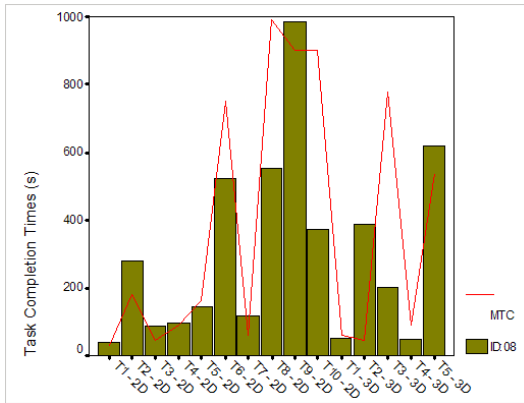
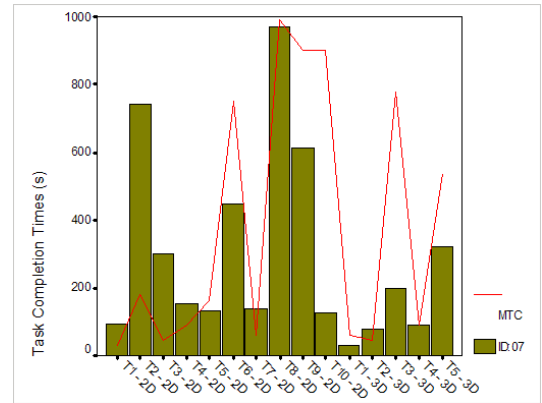
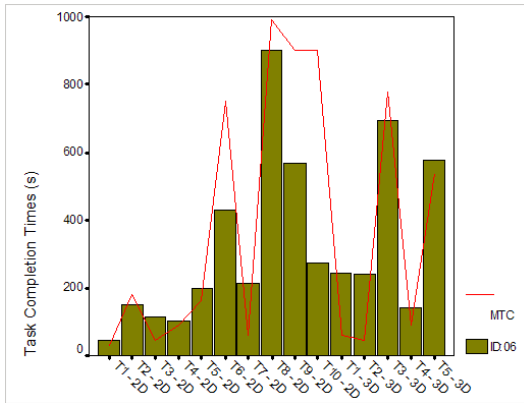
D.3 Task completion times

D.3.1 Mean task completion times



D.3.2 Task completion times for each user





Appendix E

Documents for final evaluation

E.1 User Instruction Sheet

Instructions for User Evaluation

User ID:

Thank you for agreeing to participate in this second evaluation of the two anatomy browsers.

All data obtained in the evaluation is confidential.

Although it would be helpful to allow us to get back to you with any additional questions we may have, you are welcome to omit your name and contact details if you would prefer the data to remain anonymous.

You can withdraw from the evaluation and request that your data be destroyed at any stage.

All data storage will comply with the appropriate Data Protection regulations.

At the end of the evaluation process all users will be provided with feedback on the results unless otherwise requested.

You will be given a brief overview of the browsers after completing this form. You will then be provided with a set of Task Scenario Sheets. Please use these with the help of the printout of the Quick Guide to complete the tasks detailed. Where required write out responses to questions asked on the Task Scenario sheets.

More detailed help is provided from the Help Menu in each browser. You may ask the evaluator for clarification where necessary, and as much assistance as necessary will be given provided doing so does not bias the results of the evaluation.

The evaluator will make notes on the path(s) you take to your solution for each task. A talk-through of the process you follow to achieve each goal would be appreciated as it provides more information on your understanding of how the systems work.

Once you have completed the tasks you will be provided with a brief exercise on visual information analysis, followed by two questionnaires to gather your impressions on the functionality of the system. You are welcome at this stage to provide any further information on the system that you feel has not been addressed sufficiently in the questionnaires.

If you agree to being contacted for any additional information, please tick here: ☐

If you would like to receive feedback at the end of the analysis, please tick here: ☐

If you understand and accept the above, please sign below.

Signature:

Name:

Date:

E.2 Quick guide to visualisation browsers

Menu options

File, **Edit**, **View** and **Help** menus on the menu bar (2D and 3D)

Node, **Link**, **Selection** and **Graph** popup menus in 2D only (additional popup for secondary window - *zoom pane*). Options on the popup menu correspond to those on the menu bar, but apply only to the set of nodes they describe.

Toolbar options

Options in both browsers are for **Open/Load** ontologies, **Close (All)**, **Search**, **Save State** and **Help**.

The 2D browser also provides buttons for **Zoom** and **Switch Graph Layout**, and the 3D **Set Node/Link Colour** and a link to the **Create/Draw Mappings** dialog.

Creating groups

A *group node* can be created based on user-defined criteria, using a custom dialog. Links are then drawn from the *group node* to other nodes that form a part of this group, distinguishing between child and parent nodes. Once created a *group* may be edited or removed from the graph.

Tracing lineage

Lineage paths within a single ontology in 2D trace successive component parts of a node. Lineage for a single node across multiple ontologies, describing evolution with time, may be automatically retrieved and drawn using links across trees in the 3D browser.

Options for editing graph structure

These include **Collapse/expand sub-trees**, **Ghost/highlight nodes** and **Annotate nodes and mappings** for both 2D and 3D. For 2D only there is additionally **Set number of levels to draw** and **Set label property/hide labels**.

Supplementary **textual detail** may be displayed for selected node(s) and/or link(s), and a selection of nodes may also be **extracted to a sub-window** for analysis in isolation.

Creating / drawing mappings

Available only in 3D this provides a dialog for creating mappings between any node pair loaded in the window, and allows previously created mappings to be loaded into the browser. The dialog also provides simple graphical querying to retrieve and draw specified mappings to the 3D window.

Saving to file

Options include saving a (reloadable) system state (XML) file describing graph structure and physical attributes of nodes, and in 3D only, mappings between nodes.

The graph with the focus in 2D or the current view in 3D may also be saved to a JPEG image.

E.3 Task scenario sheets

Text Indices

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
1	Display the text index for Theiler Stage (TS) 10.	REQ: working EMAP browser SCC: Text index, (2D slice and 3D model) for TS10 displayed in web browser. MTC: 34s	N/A
2	Locate the component <i>visceral endoderm</i> and determine its <i>print name</i> (fully qualified name or path to root).	REQ: working EMAP browser SCC: entry in index identified as required, correct trace to root MTC: 100s	<i>extraembryonic component.endoderm.visceral endoderm</i>
3	Identify also the component <i>visceral endoderm</i> in TS11 and TS12. Name all other components in each stage that along with the <i>visceral endoderm</i> form their complete parent component. Based on the information retrieved can you determine if this is the same component persisting through the three stages? What is the latest stage in which the <i>visceral endoderm</i> may be found?	REQ: working EMAP browser SCC: identification of the <i>visceral endoderm</i> in TS11 and TS12. MTC: 120s/30s	For all three stages - parent component: <i>extraembryonic component.endoderm</i> ; other sub-parts: <i>parietal endoderm.intermediate endoderm</i> additionally found in TS10 and TS11. GUESS TS13 because this stage does not contain the component <i>extraembryonic component.visceral endoderm</i>

2D Browser

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
1	Load Theiler Stages (TS) 10, 11 and 12 in the 2D browser.	REQ: 2D anatomy browser, Help files SCC: Graphs of TS10, 11 and 12 displayed in browser.	N/A
2	Identify the anatomy component <i>amniotic cavity</i> in TS10 and determine its <i>print name</i> (fully qualified name). Use the visual structures to determine if the components with the same name in TS11 and TS12 trace the same path to the root. Do you have enough information to tell whether or not the three nodes refer to the same structure in each stage of development?	REQ: 2D anatomy browser, Help files SCC: correct identification of node representing the <i>amniotic cavity</i> . Manual trace to determine <i>print names</i> of components in other stages.	<i>extraembryonic component.cavities.amniotic cavity</i> . Identical <i>print name</i> (paths) for TS11 and 12.
3	During which of the three stages TS10, 11 and 12, do the components <i>future brain</i> and <i>future spinal cord</i> develop, and to which stage do they persist? Of which component do they form sub-parts?	REQ: 2D anatomy browser, Help files SCC: text search or visual scan to identify components.	TS11 - from <i>embryo.ectoderm.neural ectoderm</i>
4	Which part of the <i>endoderm</i> is also referred to as the <i>hypoblast</i> ? Create a <i>group</i> in TS11 that brings together all the components that make up the <i>endoderm</i> .	REQ: 2D anatomy browser, Help files SCC: search on synonyms or identify from graph by switching label property to <i>synonym</i> . creation of group as required.	<i>hypoblast</i> is a synonym for <i>primitive endoderm</i> Group should include components with IDs: 160, 161, 162, 190, 191, 192, 194, 203. 160, 190 should be set as parents.

3D Browser

	TASK DESCRIPTION	TASK DETAIL	SOLUTIONS
1	Switch to the 3D browser (and (re)load the current structures drawn for TS10, 11 and 12). Load also TS13 and TS14.	REQ: 3D anatomy browser, Help files SCC: Graphs of TS10-14 displayed in browser.	N/A
2	Trace lineage paths from the components <i>amniotic cavity</i> in TS10 and <i>future brain</i> in TS11 respectively. Does the trace drawn in the 3D browser for the <i>future brain</i> confirm your conclusion for Task 3 for the 2D browser? Capture the current view in the 3D browser to an image file.	REQ: 3D anatomy browser, Help files SCC: lineage traces as required.	
3	Unload all stages currently drawn to the 3D window. Carnegie Stage 08 (CS08) is the equivalent stage in the development of the <i>human embryo</i> for TS11 in the <i>mouse embryo</i> . Determine the sub-parts of the component <i>embryo.ectoderm</i> in each of CS08 and TS11, and create and draw <i>homology</i> (lineage) mappings between each corresponding node pair. Take a snapshot of the system state at this point.	REQ: 3D anatomy browser, Help files SCC: correct identification of nodes. correct creation of mappings and links drawn to represent them.	Derivatives in TS11: <i>surface ectoderm</i> and <i>neural ectoderm</i> Derivatives in CS08: <i>neural ectoderm</i> and <i>non-neural ectoderm</i> (one) mapping only between components with identical <i>print names</i> .
4	Unload all stages currently drawn to the 3D window. Load the equivalent stages CS14 and TS17. Identify the two parts of the <i>limb</i> in each stage and create and draw <i>cell type</i> mappings between them. Load the file <i>mappings.xml</i> containing previously determined mappings. Using the GUI provided query the data set to retrieve and draw, alternately, ALL mappings stored in the system and only mappings for <i>analogy</i> (function) and <i>tissue type</i> .	REQ: 3D anatomy browser, Help files SCC: correct creation of mappings, and links drawn to represent them. successful querying saving system to state.	Sub-parts of <i>limb</i> for TS17: ID: 2100 - <i>hindlimb bud</i> and ID: 2096 - <i>forelimb bud</i> Sub-parts of <i>limb</i> for CS14: ID: 3193 - <i>upper limb bud</i> and ID: 3190 - <i>lower limb bud</i>
5	Switch back to the 2D layout and capture the graph in the topmost frame to an image file.	REQ: 2D & 3D anatomy browsers, Help files SCC: switch to 2D browser and saving graph to JPEG image.	N/A

E.4 Post-evaluation questionnaire

Usability Evaluation Questionnaire^a

Identification number:

Age:

Sex: ☐ Female ☐ Male

Part 1: Type of System being Rated

1.1 Did you feel you had sufficient time to familiarise yourself with the new system?

Yes ☐

No ☐

1.2 On average, how much time would you spend per week on this system?

Less than 1 hour ☐

1 to less than 4 hours ☐

4 to less than 10 hours ☐

Over 10 hours ☐

Part 2: Past Experience

2.1 Of the following devices, software, and systems, check those that you have personally used and are familiar with.

Keyboard ☐

Numeric key pad ☐

Mouse ☐

Light pen ☐

Touch screen ☐

Track ball ☐

Joy stick ☐

Text editor ☐

Word processor ☐

File manager ☐

Electronic spreadsheet ☐

Electronic mail ☐

Graphics software ☐

Computer games ☐

Colour monitor ☐

Time-share system ☐

Workstation ☐

Personal computer ☐

Floppy drive ☐

Hard drive ☐

Compact disk drive ☐

cont'd on next page

^aCopyright©1988, 1989, 1991 Human-Computer Interaction Laboratory, University of Maryland. All rights reserved. The original (Shneiderman) Usability Evaluation questionnaire has been adapted to suit this evaluation.

For sections 3-7 of the questionnaire please circle the numbers (on the scale from 1-9) which most appropriately reflect your impressions about using this computer system. Not Applicable = N/A. There is room on the last page for your written comments. (You may also make comments specific to any question in the margin to its right.)

Part 3: Overall reactions to the system										
3.1	Terrible									Wonderful
	1	2	3	4	5	6	7	8	9	N/A
3.2	Frustrating								Satisfying	
	1	2	3	4	5	6	7	8	9	N/A
3.3	Dull								Stimulating	
	1	2	3	4	5	6	7	8	9	N/A
3.4	Difficult								Easy	
	1	2	3	4	5	6	7	8	9	N/A
3.5	Inadequate power								Adequate power	
	1	2	3	4	5	6	7	8	9	N/A
3.6	Rigid								Flexible	
	1	2	3	4	5	6	7	8	9	N/A

cont'd on next page

Part 4: Data Visualisation & Screen

4.1 How representative are the visualisations generated of your mental model of the data structure?

Not at all

Very much so

1 2 3 4 5 6 7 8 9 N/A

4.2 Do you find that the visualisations of the data provide an advantage over the textual indices in the EMAP browsers?

Not at all

Very much so

1 2 3 4 5 6 7 8 9 N/A

For each of the options 4.3-4.7, compare ease of use of the visualisations generated to the EMAP text indices:

4.3 Data structure

Text indices easier to use

Visualisations easier to use

1 2 3 4 5 6 7 8 9 N/A

4.4 Understanding of data

Text indices easier to use

Visualisations easier to use

1 2 3 4 5 6 7 8 9 N/A

4.5 Search and query

Text indices easier to use

Visualisations easier to use

1 2 3 4 5 6 7 8 9 N/A

4.6 Tracing lineage

Text indices easier to use

Visualisations easier to use

1 2 3 4 5 6 7 8 9 N/A

4.7 Grouping of data

Text indices easier to use

Visualisations easier to use

1 2 3 4 5 6 7 8 9 N/A

4.8 How useful was ordering of components in the graphs?

Aided location of components

Did not aid location of components

1 2 3 4 5 6 7 8 9 N/A

4.9 Was ordering of components easier to follow in 2D or 3D?

2D ordering more useful

3D ordering more useful

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

4.10 Are the visualisations able to provide a good overview of data structure?	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A
4.11 Do the visualisations reflect your understanding of data structure?	Map to semantic content								Do not map to semantic content	
	1	2	3	4	5	6	7	8	9	N/A
4.12 Does functionality provided for detailed analysis of regions of interest overcome difficulty posed by occlusion in the overview?	Improved analysis								No improvement in analysis	
	1	2	3	4	5	6	7	8	9	N/A
4.13 How easy was it to navigate through the visual structures?	Difficult								Intuitive	
	1	2	3	4	5	6	7	8	9	N/A
4.14 How intuitive is navigation using the text indices, compared to the visual structures?	Text indices more intuitive								Visual structures more intuitive	
	1	2	3	4	5	6	7	8	9	N/A
4.15 Compare intuitiveness of navigation through the data structures in 2D and 3D.	2D more intuitive								3D more intuitive	
	1	2	3	4	5	6	7	8	9	N/A
For questions 4.16-4.20 please indicate the level of usefulness of each of the options available in 2D for the reduction of occlusion (or the option N/A for those functions not used.)										
4.16 Hiding of labels	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A
4.17 Ghosting of data	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A
4.18 Hiding of sub-trees	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A
4.19 Zoom	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A
4.20 Switching between layouts	Useful								Not useful	
	1	2	3	4	5	6	7	8	9	N/A

cont'd on next page

For questions 4.21-4.22 please indicate the level of usefulness of each of the options available in 3D for the reduction of occlusion (or the option N/A for those functions not used.)

4.21 Hiding of sub-trees

Useful										Not useful	
1		2	3	4	5	6	7	8		9	N/A

4.22 Zoom

Useful										Not useful	
1		2	3	4	5	6	7	8		9	N/A

4.23 How easy was it to locate information required?

Difficult to find										Easy to find	
1		2	3	4	5	6	7	8		9	N/A

4.24 How easy was it to perform search and query operations?

Difficult										Easy	
1		2	3	4	5	6	7	8		9	N/A

4.25 How intuitive is interpretation of search and query results?

Difficult										Easy	
1		2	3	4	5	6	7	8		9	N/A

4.26 Do the visualisations ease determination of lineage, compared to the system currently in place?

Less intuitive										More intuitive	
1		2	3	4	5	6	7	8		9	N/A

4.27 How useful is the 2D browser for tracing lineage, compared to the 3D?

3D more intuitive										2D more intuitive	
1		2	3	4	5	6	7	8		9	N/A

4.28 How well does the graphical support provided for grouping of data highlight user-specified data groups?

Poorly highlighted										Well highlighted	
1		2	3	4	5	6	7	8		9	N/A

4.29 How useful is grouping in the 2D browser compared to the 3D?

2D layout well highlighted										3D layout well highlighted	
1		2	3	4	5	6	7	8		9	N/A

cont'd on next page

Part 5: Terminology & System Information

5.1 Use of terms throughout system

Inconsistent

Consistent

1 2 3 4 5 6 7 8 9 N/A

5.2 Does the terminology relate well to the work you are doing?

Unrelated

Well related

1 2 3 4 5 6 7 8 9 N/A

5.3 Messages which appear on screen

Inconsistent

Consistent

1 2 3 4 5 6 7 8 9 N/A

5.4 Messages which appear on screen

Confusing

Clear

1 2 3 4 5 6 7 8 9 N/A

5.5 Does the computer keep you informed about what it is doing?

Never

Always

1 2 3 4 5 6 7 8 9 N/A

5.6 Error messages

Unhelpful

Helpful

1 2 3 4 5 6 7 8 9 N/A

5.7 System feedback

Unhelpful

Helpful

1 2 3 4 5 6 7 8 9 N/A

5.8 Ability to identify errors and sources of errors

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

5.9 System help/support

Not useful

Useful

1 2 3 4 5 6 7 8 9 N/A

5.10 Level of system support for error recovery

Low

High

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

Part 6: Learning

6.1 Understanding of terms used throughout system

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

6.2 Understanding of messages which appear on screen

Ambiguous

Unambiguous

1 2 3 4 5 6 7 8 9 N/A

6.3 Usefulness of messages which appear on screen

Never

Always

1 2 3 4 5 6 7 8 9 N/A

6.4 Error messages

Confusing

Clear

1 2 3 4 5 6 7 8 9 N/A

6.5 Does the computer keep you informed about what it is doing?

Confusing

Clear

1 2 3 4 5 6 7 8 9 N/A

6.6 Ease of learning of the functions available in the visualisation browsers

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

6.7 Usefulness of functionality provided for querying data

Useful

Not useful

1 2 3 4 5 6 7 8 9 N/A

6.8 Compare ease of learning of the functions available in 2D and 3D

2D more intuitive

3D more intuitive

1 2 3 4 5 6 7 8 9 N/A

Please rate actual ability to make use of the 2D browser (6.9-6.15).

6.9 Navigation through data

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

6.10 Location of specific information required

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

6.11 Understanding of data structure

Difficult

Easy

1 2 3 4 5 6 7 8 9 N/A

cont'd on next page

6.12 Understanding of data encoding										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.13 Querying data for information required										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.14 Understanding of visual query results										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.15 Usefulness of visual cues provided for querying										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
Please rate actual ability to make use of the 3D browser (6.16-6.22).										
6.16 Navigation through the data										
	Difficult								Easy	
	1	2	3	4	5	6	7	8	9	N/A
6.17 Location of specific information required										
	Difficult								Easy	
	1	2	3	4	5	6	7	8	9	N/A
6.18 Understanding of data structure										
	Difficult								Easy	
	1	2	3	4	5	6	7	8	9	N/A
6.19 Understanding of data encoding										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.20 Querying data for information required										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.21 Understanding of visual query results										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A
6.22 Usefulness of visual cues provided for querying										
	Intuitive								Non-intuitive	
	1	2	3	4	5	6	7	8	9	N/A

cont'd on next page

6.23 How useful are textual query results in isolation in either browser?									
	Useful				Not useful				
	1	2	3	4	5	6	7	8	9
6.24 How easy is it to understand textual query results?									N/A
	Difficult				Easy				
	1	2	3	4	5	6	7	8	9
6.25 How useful are visual query results in isolation?									N/A
	Useful				Not useful				
	1	2	3	4	5	6	7	8	9
6.26 How easy is it to locate specific components by tracking location based on path to the root in each graph?									N/A
	Difficult				Easy				
	1	2	3	4	5	6	7	8	9
6.27 How easy is it to locate specific components using the search dialog?									N/A
	Difficult				Easy				
	1	2	3	4	5	6	7	8	9
6.28 Compare querying using the EMAP text indices to use of the visualisations.									N/A
	Text indices more intuitive				Visual structures more intuitive				
	1	2	3	4	5	6	7	8	9
6.29 Compare functionality for querying in the text indices to that provided for the visualisations.									N/A
	Text indices more useful				Visual structures more useful				
	1	2	3	4	5	6	7	8	9
6.30 Compare querying using the 2D browser to the 3D.									N/A
	2D browser more intuitive				3D browser more intuitive				
	1	2	3	4	5	6	7	8	9
6.31 How would you rate the time you required, on average, to perform tasks?									N/A
	Very long				Very short				
	1	2	3	4	5	6	7	8	9
6.32 How long do you feel you would require to reach a working level of proficiency?									
	> 1 year	1 year	6 mths	1 mth	2 weeks				1 day

cont'd on next page

Part 7: System Capabilities

7.1 System speed, on average

Too slow**Fast enough**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.2 Variations in system speed

Large**Small**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.3 Correcting your mistakes

Difficult**Easy**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.4 Are the needs of both experienced and inexperienced users taken into account?

Never**Always**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.5 How would you rate the level of functionality offered by the system?

Poor**Very good**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

Compared to the current working browsers how would you rate this system:

7.6

Difficult to use**Easy to use**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.7

**Difficult data
analysis****Simplified data
analysis**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

7.8

Unintuitive**Intuitive**

1	2	3	4	5	6	7	8	9	N/A
---	---	---	---	---	---	---	---	---	-----

cont'd on next page

For questions 7.9-7.16 compare use of the 2D browser to the 3D. (Choose the middle point to indicate no preference/advantage of one system over the other.)

7.9 Navigation through visual structures

2D more intuitive

1 2 3 4 5 6 7 8

3D more intuitive

9

7.10 Data analysis

More difficult in 2D

1 2 3 4 5 6 7 8

More difficult in 3D

9

7.11 Ability of visualisations to provide an overview of data structure

2D more useful

1 2 3 4 5 6 7 8

3D more useful

9

7.12 Usefulness of visual structures for analysis

2D more intuitive

1 2 3 4 5 6 7 8

3D more intuitive

9

7.13 Locating data of interest

2D more effective

1 2 3 4 5 6 7 8

3D more effective

9

7.14 Identifying relationships in data

2D more intuitive

1 2 3 4 5 6 7 8

3D more intuitive

9

7.15 Support for creating and displaying groups

2D more effective

1 2 3 4 5 6 7 8

3D more effective

9

7.16 Functionality for tracing lineage

2D more intuitive

1 2 3 4 5 6 7 8

3D more intuitive

9

7.17 Rate system response in the 2D browser

Good response

1 2 3 4 5 6 7 8

Poor response

9

N/A

7.18 Rate system response in the 3D browser

Good response

1 2 3 4 5 6 7 8

Poor response

9

N/A

cont'd on next page

Part 8: Functionality

Please list, in decreasing order of usefulness, cues you identified in the browsers that improve data analysis and information retrieval, indicating what you used them for.

1. _____

2. _____

3. _____

4. _____

5. _____

Part 9: Use of EMAP browsers

9.1 How frequently do you use the currently working EMAP (Mouse Atlas) browsers?

Never Occasionally Monthly Weekly Daily

9.2 For how long have you been using the EMAP browsers?

< 1 mth 1-3 mths 3-6 mths 6 mths-1yr > 1 yr N/A

cont'd on next page

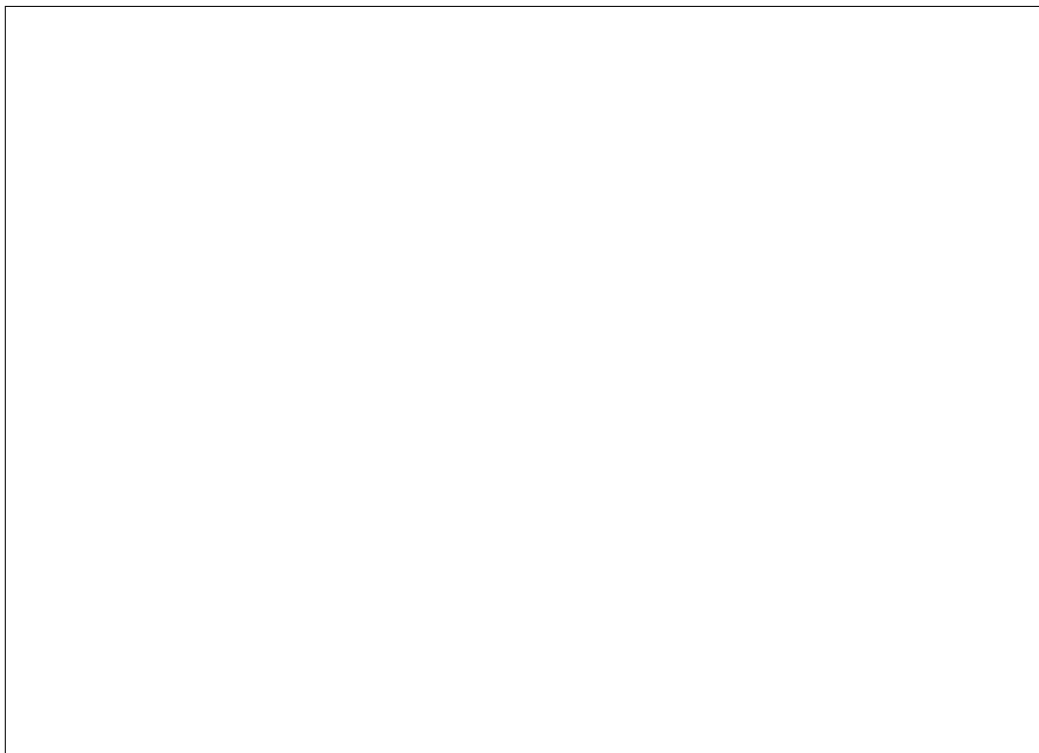

Part 10: Users' Comments

Please write any comments you have in the space below.

E.5 Spatial ability/awareness exercises

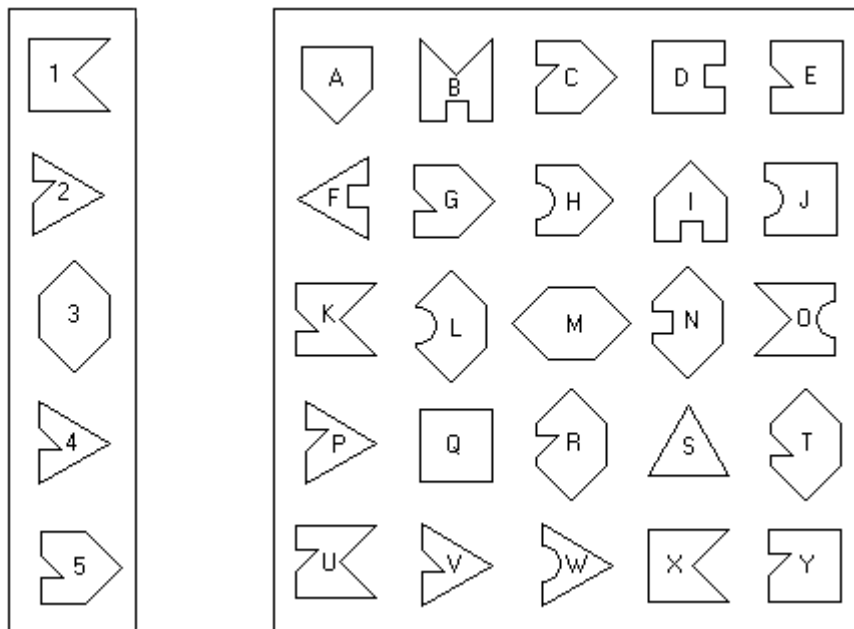
Exercise 1

Please sketch, for each of the 2D and 3D browsers, your understanding/recollection of the visual structure for which you created a group.

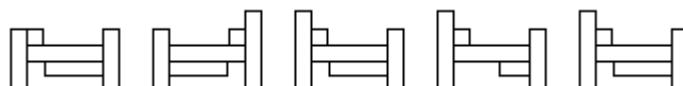


Exercise 2^a

- Choose, for each of the shapes 1-5 the corresponding shape from the options A-Y.

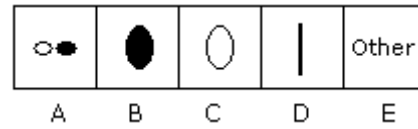
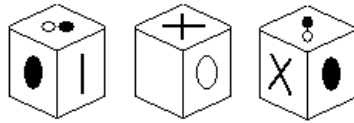


- Choose the two identical shapes out of the five shown.



^aSample exercises Copyright©Psychometric Success - available at:
<http://www.psychometric-success.com/Aptitude/%20Tests/%207.htm>

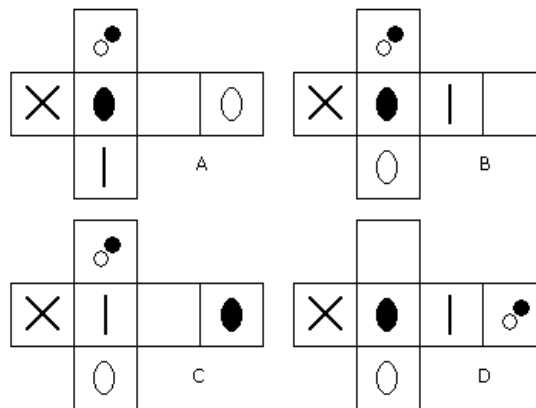
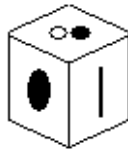
3. Which of the options on the right is opposite the figure X on the cube shown on the left?



4. Which of the figures in the group on the right is a rotation of the one on the left?



5. Which of the options below can be folded to obtain the cube shown?



Exercise 3^a

Choose from the options on the right the next in the sequence on the left-hand side.

1.		 a b c d e
2.		 a b c d e
3.		 a b c d e
4.		 a b c d e
5.		 a b c d e
6.		 a b c d e
7.		 a b c d e
8.		 a b c d e

^aSample exercises Copyright©SHL Group plc,1998 - available at:
<http://www.shldirect.com/phaseI/helpsection-phaseI/Help-on-16.asp?ID=EBEE26F9306A4A82B5EC8C6283DBBD1>

E.6 Sample log recording use of browsers

Entry time - 08:54:46		Group Nodes - 09:00:09	09:00:38
Open - 08:54:50		Switch to 3D - 09:01:04	
Open - 08:54:55		Open - 09:01:20	09:01:43
Open - 08:55:00	08:55:03		
		Search - 09:02:13	
Set Max Levels - 08:55:18		Auto Lineage - 09:02:46	09:03:00
TS10 - 08:55:20			
max node count: 42 - 08:55:20	08:55:20	Auto Lineage - 09:03:32	09:04:31
Search - 08:55:22	08:55:58	Close All - 09:04:49	
		Open - 09:05:15	
Set Max Levels - 08:56:14		Open - 09:05:22	
TS11 - 08:56:16		Search - 09:05:46	09:06:07
max node count: 61 - 08:56:16			
08:56:16		Draw Selected Mappings - 09:07:34	09:07:48
Set Max Levels - 08:56:28			
TS12 - 08:56:30		Save State - 09:09:16	09:09:25
max node count: 199 - 08:56:30	08:56:30	Close All - 09:09:39	
		Open - 09:09:49	
TS12 - 08:56:36		Open - 09:09:59	09:10:27
max node count: 199 - 08:56:36	08:57:04		
		Search - 09:10:43	09:13:55
Search - 08:57:21			
Search - 08:57:32	08:57:51	Load Mappings - 09:14:11	09:14:21
Show Selection Detail - 08:58:09	08:58:24	Draw All Mappings - 09:14:51	
TS12 - 08:58:49		Draw All Mappings - 09:15:17	
max node count: 199 - 08:58:49	08:58:55	Save Image - 09:15:42	
		Switch to 2D - 09:16:13	
Search - 08:59:16	08:59:41		
		Exit time - 09:16:21	

Appendix F

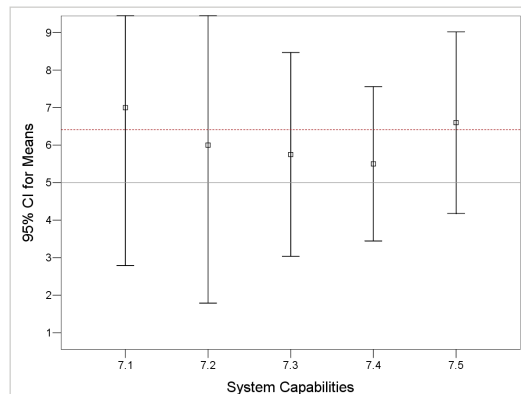
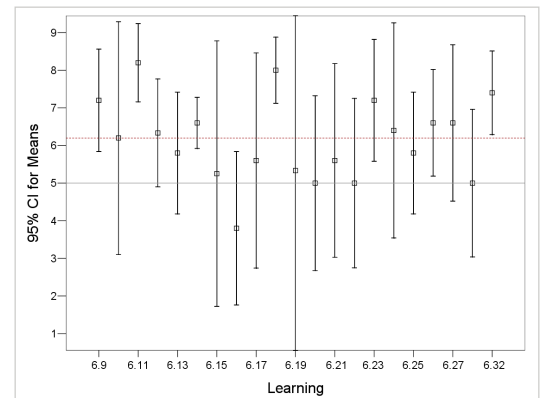
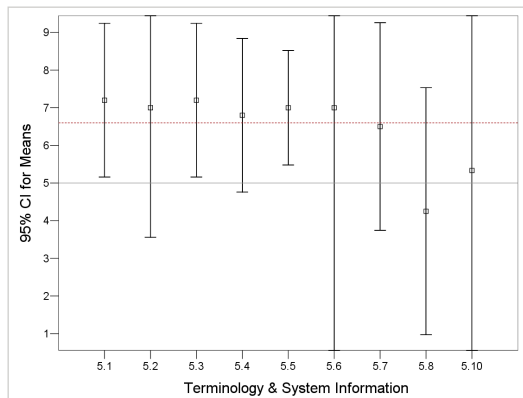
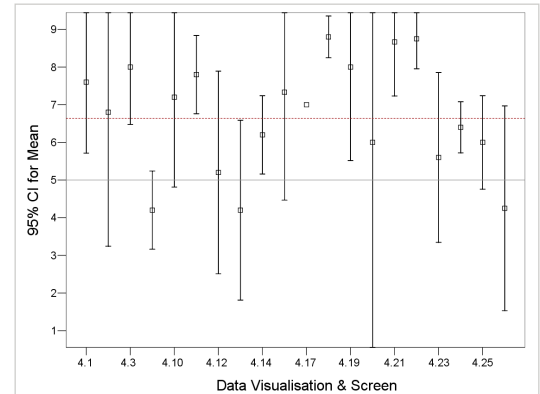
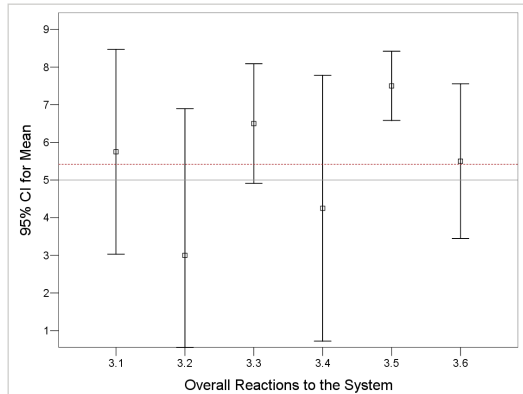
Results for final evaluation

F.1 Range of specifications for users' computers

O/S	CPU	Memory	Hard drive	Monitor
Windows XP	P4 1.7GHz	256MB	20Gb	768 * 1024
Windows XP	P4 2GHz	1GB	80Gb	1280 * 1024
	AT/AT Compatible			
Linux Redhat	P4 CPU 2.40GHz	1GB	80Gb	768 * 1024

F.2 Post-evaluation questionnaire

F.2.1 Mean usability satisfaction rankings



References

- [1] F. Achard, G. Vaysseix, and E. Barillot, “XML, bioinformatics and data integration,” *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 2, pp. 115–125.
- [2] P. Adriaans and D. Zantinge, *Data mining*. Addison-Wesley, 1996, (176 pps).
- [3] C. Ahlberg, C. Williamson, and B. Shneiderman, “Dynamic queries for information exploration: an implementation and evaluation,” in *CHI '92: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, P. Bauersfeld, J. Bennett, and G. Lynch, Eds. ACM Press, 1992, pp. 619–626.
- [4] H. Alani, “TGVizTab: An ontology visualisation extension for Protégé,” in *Visualizing Information in Knowledge Engineering (VIKE '03), A workshop at the 2nd International Conference on Knowledge CAPture Proceedings of Knowledge Capture (K-Cap'03)*. ACM Press, 2003, (6 pps).
- [5] C. Alberola, L. Tardon, and J. Ruiz-Alzola, “Graphical models for problem solving,” *Computing in Science & Engineering*. IEEE Computer Society and the American Institute of Physics, 2000, vol. 2, no. 4, pp. 46–57.
- [6] B. Allen, “Information space representation in interactive systems: relationship to spatial abilities,” in *Proceedings of the third ACM conference on Digital Libraries*. ACM Press, 1998, pp. 1–10.
- [7] G. E. Allen, “Essays on science and society: Is a new eugenics afoot?” *Science*. AAAS, 2001, vol. 294, no. 5540, pp. 59–61.
- [8] J. Allen, “In silico veritas. data-mining and automated discovery: the truth is in there.” *EMBO Reports*. Nature Publishing Group, 2001, vol. 2, no. 7, pp. 542–544.
- [9] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, “Gene ontology: tool for the unification of biology,” *Nature Genetics*. Nature Publishing Group, 2000, vol. 25, pp. 25–29.
- [10] T. K. Attwood and D. J. Parry-Smith, *Introduction to Bioinformatics*. Addison-Wesley Longman, 1999, (240 pps).
- [11] E. H. Baehrecke, N. Dang, K. Babaria, and B. Shneiderman, “Visualization and analysis of microarray and gene ontology data with treemaps,” *BMC Bioinformatics*. BioMed Central, 2004, vol. 5, no. 84, (12 pps).

- [12] P. Baker, C. Goble, S. Bechhofer, N. Paton, R. Stevens, and A. Brass, “An ontology for bioinformatics applications,” *Bioinformatics*. Oxford University Press, 1999, vol. 15, no. 6, pp. 510–520.
- [13] R. Baldock, C. Dubreuil, W. Hill, and D. Davidson, “The Edinburgh Mouse Atlas: Basic structure and informatics,” in *Bioinformatics: databases and systems*, S. Letovsky, Ed. Kluwer Academic Publishers, 1999, pp. 129–140.
- [14] R. Baldock and A. Burger, “Anatomical ontologies: names and places in biology,” *Genome Biology*. BioMed Central, 2005, vol. 6, no. 108, (12 pps).
- [15] R. A. Baldock, J. B. L. Bard, A. Burger, N. Burton, J. Christiansen, G. Feng, B. Hill, D. Houghton, M. Kaufman, J. Rao, J. Sharpe, A. Ross, P. Stevenson, S. Venkataraman, A. Waterhouse, Y. Yang, and D. R. Davidson, “EMAP and EMAGE: A framework for understanding spatially organized data,” *Neuroinformatics*. Humana Press, 2003, vol. 1, no. 4, pp. 309–326.
- [16] R. E. Barber and H. C. Lucas, Jr., “System response time operator productivity, and job satisfaction,” *Communications of the ACM*. ACM Press, 1983, vol. 26, no. 11, pp. 972–986.
- [17] J. B. Bard, R. A. Baldock, and D. R. Davidson, “Elucidating the genetic networks of development: A bioinformatics approach,” *Genome Research*. Cold Spring Harbor Laboratory Press, 1998, vol. 8, no. 9, pp. 859–863.
- [18] J. Barnes and J. Robertson, “The use of ontologies in drug discovery,” *Bioinformatics World*, 2002.
- [19] E. Battistella, J. G. C. de Souza, R. A. Ferreira, R. Vieira, J. C. M. Mombach, and N. Lemke, “Bioinformatics: A growing field for ontologies,” in *Workshop on Ontologies and their Applications (WONTO’2004)*, 2004, pp. 1–11.
- [20] S. Batzoglou, L. Pachter, J. Mesirov, B. Berger, and E. S. Lander, “Human and mouse gene structure: comparative analysis and application to exon prediction,” in *RECOMB ’00: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, R. Shamir, S. Miyano, S. Istrail, P. Pevzner, and M. Waterman, Eds. ACM Press, 2000, pp. 46–53.
- [21] B. Bederson and A. Boltman, “Does animation help users build mental maps of spatial information?” in *Proceedings, 1999 IEEE Symposium on Information Visualization*. IEEE Computer Society, 1999, pp. 28–35.
- [22] B. Bederson, J. Grosjean, and J. Meyer, “Toolkit design for interactive structured graphics,” *IEEE Transactions on Software Engineering*. IEEE Press, 2004, vol. 30, no. 8, pp. 535–546.
- [23] D. Benton, “Bioinformatics — principles and potential of a new multidisciplinary tool,” *Trends in Biotechnology*. Elsevier Science Publishers, 1996, vol. 14, no. 8, pp. 261–272.
- [24] D. Benyon and K. Höök, “Navigation in information spaces: Supporting the individual,” in *IFIP International Conference on Human-Computer Interaction, INTERACT ’97*, S. Howard, J. Hammond, and G. Lindgaard, Eds. Chapman and Hall, 1997, pp. 39–46.
- [25] J. Bertin, *Semiology of Graphics*. University of Wisconsin Press, 1983, (432 pps), translated by W. J. Berg from J. Bertin, *Sémiologie Graphique*, 1967.

- [26] D. Brodbeck and L. Girardin, "Design study: Using multiple coordinated views to analyze geo-referenced high-dimensional datasets," in *Proceedings, International Conference on Coordinated and Multiple Views in Exploratory Visualization*, J. Roberts, Ed. IEEE Computer Society, 2003, pp. 104–111.
- [27] J. Brooke, "SUS: A 'quick and dirty' usability scale," in *Usability Evaluation in Industry*, P. Jordan, B. Thomas, B. Weerdmeester, and I. McClelland, Eds. Taylor & Francis, 1996, pp. 189–194.
- [28] R. Brune, J. Bard, C. Dubreuil, E. Guest, W. Hill, M. Kaufman, M. Stark, D. Davidson, and R. Baldock, "A three-dimensional model of the mouse at embryonic day 9," *Developmental Biology*. Academic Press, 1999, vol. 216, no. 2, pp. 457–468.
- [29] C. Bult, J. Richardson, J. Blake, J. Kadin, M. Ringwald, J. Eppig, R. Baldarelli, M. Baya, J. Beal, D. Begley, W. Boddy, D. Bradt, N. Butler, T. Chu, L. Corbani, J. Corradi, M. Davisson, D. Garippa, L. Glass, P. Grant, D. Hill, B. King, D. Krupke, M. Lennon-Pierce, C. Lutz, L. Maltais, P. Mani, I. McCright, L. McKenzie, D. Naf, J. Ormsby, S. Ramachandran, D. Reed, D. Shaw, P. Szauter, and L. Trombley, "Mouse genome informatics in a new age of biological inquiry," in *Proceedings, IEEE International Symposium on Bio-Informatics and Biomedical Engineering*. IEEE Computer Society, 2000, pp. 29–32.
- [30] A. Burger, R. Baldock, Y. Yang, A. Waterhouse, D. Houghton, N. Burton, and D. Davidson, "Poster: The Edinburgh Mouse Atlas and Gene-Expression Database: a spatio-temporal database for biological research," in *Proceedings, 14th International Conference on Scientific and Statistical Database Management, 2002*, J. Kennedy, Ed. IEEE Computer Society, 2002, p. 239.
- [31] A. Burger, B. Webber, W. Nutt, S. Wagner, S. Aitken, G. Ferguson, P. Holt, and J. Bard, "(abstract) XSPAN: A Cross-Species Anatomy Network," in *Standards and Ontologies for Functional Genomics (SOFG)*, 2002, p. 9.
- [32] A. Burger, "A systematic nomenclature for embryo anatomy," in *Manchester Bioinformatics Week: Ontology Workshop*, 2002.
- [33] A. Burger, D. Davidson, and R. Baldock, "Formalization of mouse embryo anatomy," *Bioinformatics*. Oxford University Press, 2004, vol. 20, no. 2, pp. 259–267.
- [34] A. Burger, D. Davidson, Y. Yang, and R. Baldock, "Integrating partonomic hierarchies in anatomy ontologies," *BMC Bioinformatics*. BioMed Central, 2004, vol. 5, no. 184, (10 pps).
- [35] C. Burton and L. Johnston, "Will World Wide Web user interfaces be usable?" in *Proceedings, 1998 Australasian Computer Human Interaction Conference. OzCHI'98*, P. Calder and B. Thomas, Eds., Ergonomics Society of Australia. IEEE Computer Society, 1998, pp. 39–44.
- [36] D. Butler, "Are you ready for the revolution?" *Nature*. Nature Publishing Group, 2001, vol. 409, pp. 758–760.
- [37] S. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualisation: Using Vision to Think*. Morgan Kaufmann Publishers Inc, 1999, (686 pps).
- [38] T. Catarci, "Databases and the Web: New requirements for easy access," *ACM Computing Surveys* 28(4es), *Special issue: position statements on strategic directions in computing research*. ACM Press, 1996, vol. 28, no. 4es, Article 131, (3 pps).

- [39] M. Chalmers, "Design perspectives in visualising complex information," in *Visual Database Systems 3, Visual Information Management, Proceedings of the third IFIP working conference on visual database systems*, S. Spaccapietra and R. Jain, Eds. Chapman Hall, 1995, pp. 103–111.
- [40] M. Chalmers, R. Ingram, and C. Pfranger, "Adding imageability features to information displays," in *UIST '96: Proceedings of the 9th annual ACM symposium on User Interface Software and Technology*. ACM Press, 1996, pp. 33–39.
- [41] J. M. Chambers, W. Cleveland, B. Kleiner, and P. Tukey, *Graphical Methods for Data Analysis*. Wadsworth and Brooks/Cole Publishing, 1983, (330 pps).
- [42] B. Chandrasekaran, J. Josephson, and V. Benjamins, "What are ontologies, and why do we need them?" *IEEE Intelligent Systems and Their Applications*. IEEE Computer Society, 1999, vol. 14, no. 1, pp. 20–26.
- [43] M. Chapman and C. Wykes, *Plain figures*, 2nd ed. London: The Stationery Office, 1996, (148 pps).
- [44] C. Chen and M. Czerwinski, "Spatial ability and visual navigation: An empirical study," *The New Review of Hypermedia and Multimedia*. Taylor & Francis, 1997, vol. 3, pp. 67–89.
- [45] E. S. Chen and D. B. Davison, "Distributing molecular biology information: Gopher, WAIS and the University of Houston Gene-Server," in *Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing*, E. Deaton, K. M. George, H. Berghel, and G. Hedrick, Eds. ACM Press, 1993, pp. 634–640.
- [46] K.-H. Cheung and D.-G. Shin, "A graph-based meta-data framework for interoperation between genome databases," in *Proceedings, IEEE International Symposium on Bio-Informatics and Biomedical Engineering, 2000*. IEEE Computer Society, 2000, pp. 109–117.
- [47] M. Chuah and S. Roth, "On the semantics of interactive visualizations," in *Proceedings, IEEE Symposium on Information Visualization '96*, S. Card, S. G. Eick, and N. Gershon, Eds. IEEE Computer Society, 1996, pp. 29–36.
- [48] M. C. Chuah, S. F. Roth, J. Mattis, and J. Kolojejchick, "SDM: selective dynamic manipulation of visualizations," in *UIST '95: Proceedings of the 8th annual ACM symposium on User Interface and Software Technology*. ACM Press, 1995, pp. 61–70.
- [49] W. Cleveland, *The Elements of Graphing Data*. Wadsworth, 1985, (323 pps).
- [50] A. Cockburn and B. McKenzie, "Evaluating spatial memory in two and three dimensions," *International Journal of Human-Computer Studies*. Academic Press, 2004, vol. 61, no. 3, pp. 359–373.
- [51] A. Cockburn and B. McKenzie, "Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments," in *CHI '02: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM Press, 2002, pp. 203–210.
- [52] A.-S. Dadzie and A. Burger, "Providing visualisation support for the analysis of anatomy ontology data," *BMC Bioinformatics*. BioMed Central, 2005, vol. 6, no. 74, (21 pps).

- [53] A.-S. Dadzie and A. Burger, "The merits of the third dimension for visual analysis of multiple anatomy ontologies," in *Advances In Bioinformatics And Its Applications: Proceedings of the International Conference on Bioinformatics and Its Applications (ICBA'04)*, M. He, G. Narasimhan, and S. Petoukhov, Eds. World Scientific Publishing Company, 2005, pp. 576–587.
- [54] D. Davidson, J. Bard, M. Kaufman, and R. Baldock, "The Mouse Atlas Database: a community resource for mouse development," *Trends in Genetics*. Elsevier Science, 2001, vol. 17, no. 1, pp. 49–51.
- [55] P. N. Day, "An investigation into the cognitive effects of delayed visual feedback," Ph.D. thesis, Heriot-Watt University, Edinburgh, Scotland, 2003.
- [56] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999, (432 pps).
- [57] M. Dodge and R. Kitchin, *Atlas of Cyberspace*. Addison Wesley, 2001, (279 pps).
- [58] U. Dogrusoz, Q. Feng, B. Madden, M. Doorley, and A. Frick, "Graph visualization toolkits," *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2002, vol. 22, no. 1, pp. 30–37.
- [59] N. Drew and B. Hendley, "Visualising complex interacting systems," in *CHI '95: Conference companion on Human Factors in Computing Systems*. ACM Press, 1995, pp. 204–205.
- [60] B. Dysvik and I. Jonassen, "J-Express: exploring gene expression data using Java," *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 4, pp. 369–370.
- [61] A. Ehrenberg, "How presentation graphs communicate," in *Proceedings, IEEE International Conference on Information Visualization, 2000*, E. Banissi, M. Bannatyne, C. Chen, F. Khosrowshahi, M. Sarfraz, and A. Ursyn, Eds. IEEE Computer Society, 2000, pp. 206–212.
- [62] S. G. Eick, "Visualizing online activity," *Communications of the ACM*. ACM Press, 2001, pp. 45–50.
- [63] S. G. Eick and G. J. Wills, "Navigating large networks with hierarchies," in *Proceedings, IEEE Visualization '93*, G. M. Nielson and D. Bergeron, Eds. IEEE Computer Society, 1993, pp. 204–210.
- [64] P. Eklund, N. Roberts, and S. Green, "Ontorama: Browsing RDF ontologies using a hyperbolic-style browser," in *Proceedings, First International Symposium on Cyber Worlds, 2002*, S. Peng, V. V. Savchenko, and S. Yukita, Eds. IEEE Computer Society, 2002, pp. 405–411.
- [65] A. J. Enright and C. A. Ouzounis, "BioLayout — an automatic graph layout algorithm for similarity visualization," *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 9, pp. 853–854.
- [66] N. A. Ernst, M.-A. Storey, and P. Allen, "Cognitive support for ontology modeling," *International Journal of Human-Computer Studies*. Academic Press, 2005, vol. 62, no. 5, pp. 553–577.
- [67] J. O. Everett, D. G. Bobrow, R. Stolle, R. Crouch, V. de Paiva, C. Condoravdi, M. van den Berg, and L. Polanyi, "Making ontologies work for resolving redundancies across documents," *Communications of the ACM, Ontology: different ways of representing the same concept SPECIAL ISSUE: Ontology applications and design*. ACM Press, 2002, vol. 45, no. 2, pp. 55–60.

- [68] R. M. Ewing and J. M. Cherry, "Visualization of expression clusters using Sammon's non-linear mapping," *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 7, pp. 658–659.
- [69] G. Falkman, "Issues in structured knowledge representation: A definitional approach with application to case-based reasoning and medical informatics," Ph.D. thesis, Chalmers University of Technology and Göteborg University, Göteborg, Sweden, 2003, (225 pps).
- [70] X. Faulkner, *Usability Engineering*. Palgrave Macmillan, 2000, (256 pps).
- [71] U. Fayyad, G. G. Grinstein, and A. Wierse, Eds., *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2002, (407 pps).
- [72] U. Fayyad, D. Haussler, and P. Stolorz, "Mining scientific data," *Communications of the ACM*. ACM Press, 1996, vol. 39, no. 11, pp. 51–57.
- [73] U. Fayyad and R. Uthrusamy, "Data mining and knowledge discovery in databases," *Communications of the ACM*. ACM Press, 1996, vol. 39, no. 11, pp. 24–26.
- [74] C. Fluit, M. Sabou, and F. van Harmelen, "Ontology-based information visualisation," in *Visualizing the Semantic Web*, V. Geroimenko and C. Chen, Eds. Springer Verlag, 2002, pp. 36–48.
- [75] H. P. Frei and D. Stieger, "The use of semantic links in hypertext information retrieval," *Information Processing & Management*. Elsevier, 1995, vol. 31, no. 1, pp. 1–13.
- [76] M. Fröhlich and M. Werner, "The graph visualization system daVinci - a user interface for applications," Department of Computer Science, Universität Bremen, Tech. Rep. 5/94;:1–13, 1994, (last viewed Jul 2006). Available online: http://www.informatik.uni-bremen.de/agbkb/publikationen/bibsearch/detail.e.htm?pk_int=8
- [77] B. J. Fry, "Organic information design," Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2000, (97 pps).
- [78] B. J. Fry, "Computational information design," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2004, (170 pps).
- [79] G. W. Furnas, "The FISHEYE view: A new look at structured files," Bell Laboratories Technical Memorandum, Tech. Rep. 82-11221-2;:1–23, 1982, (last viewed Jul 2006). Available online: <http://www.si.umich.edu/~furnas/Papers/FisheyeOriginalTM.pdf>
- [80] G. W. Furnas, "Generalized fisheye views," in *CHI '86: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, M. Mantei and P. Orbeton, Eds. ACM Press, 1986, pp. 16–23.
- [81] G. W. Furnas, "New graphical reasoning models for understanding graphical interfaces," in *CHI '91: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, S. P. Robertson, G. M. Olson, and J. S. Olson, Eds. ACM Press, 1991, pp. 71–78.
- [82] G. W. Furnas, "A fisheye follow-up: further reflections on focus + context," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM Press, 2006, pp. 999–1008.
- [83] E. Gansner, E. Koutsofios, S. North, and K. Vo, "Graph visualization in software analysis," in *Proceedings of the Second Symposium on Assessment of Quality Software Development Tools, 1992*, E. Nahouraii, Ed. IEEE Computer Society Press, 1992.

- [84] W. M. Gelbart, "Databases in genomic research," *Science*. AAAS, 1998, vol. 282, pp. 659–661.
- [85] J. H. Gennari, A. Silberfein, and J. C. Wiley, "Integrating genomic knowledge sources through an anatomy ontology," in *Biocomputing 2005, Proceedings of the Pacific Symposium on Biocomputing*, R. B. Altman, T. A. Jung, T. E. Klein, A. K. Dunker, and L. Hunter, Eds. World Scientific, 2005, pp. 128–139.
- [86] N. Gershon and W. Page, "What storytelling can do for information visualization," *Communications of the ACM*. ACM Press, 2001, vol. 44, no. 8, pp. 31–37.
- [87] M. Gerstein, "Integrative database analysis in structural genomics," *Nature Structural Biology, Structural Genomics Supplement*. Nature Publishing Group, 2000, vol. 7, no. 11s, pp. 960–963.
- [88] D. Gilbert, M. Schroeder, and J. van Helden, "Interactive visualisation and exploration of biological data," in *Second International Workshop on Biomolecular Informatics in conjunction with Fifth Joint Conference on Information Sciences*, 2000, (4pps).
- [89] D. Gilbert, M. Schroeder, and J. van Helden, "Interactive visualization and exploration of relationships between biological objects," *Trends in Biotechnology*. Elsevier Science Publishers, 2000, vol. 18, no. 12, pp. 487–494.
- [90] C. Goble, "Supporting web based biology with ontologies," in *Proceedings, 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine*, S. Laxminarayanan, A. Marsh, M. Akay, I. Iakovidis, L. Kun, and C. Roux, Eds. IEEE, 2000, pp. 384–389.
- [91] M. Graham, J. Kennedy, and D. Bento, "Towards a methodology for developing visualizations," *International Journal of Human-Computer Studies*. Academic Press, 2000, vol. 53, no. 5, pp. 789–807.
- [92] H. T. Greely, "Editorial, genomics research and human subjects," *Science*. AAAS, 1998, vol. 282, no. 5389, p. 625.
- [93] M. A. Harris *et al.*, "The Gene Ontology (GO) database and informatics resource," *Nucleic Acids Research*. Oxford University Press, 2004, vol. 32, pp. D258–D261.
- [94] T. F. Hayamizu, M. Mangan, J. P. Corradi, J. A. Kadin, and M. Ringwald, "The adult mouse anatomical dictionary: a tool for annotating and integrating data," *Genome Biology*. BioMed Central, 2005, vol. 6, no. R29, (8 pps).
- [95] R. Hendley, N. Drew, A. Wood, and R. Beale, "Case study. Narcissus: visualising information," in *Proceedings, Information Visualization, 1995*, N. Gershon and S. Eick, Eds. IEEE Computer Society Press, 1995, pp. 90–96, 146.
- [96] I. Herman, M. Delest, and G. Melançon, "Tree visualisation and navigation clues for information visualisation," *Computer Graphics Forum*. Blackwell Publishing, 1998, vol. 17, no. 2, pp. 153–165.
- [97] I. Herman and D. Duke, "Minimal graphics," *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2001, vol. 21, no. 6, pp. 18–21.
- [98] I. Herman, G. Melançon, M. de Ruiter, and M. Delest, "Latour — a tree visualisation system," National Research Institute for Mathematics and Computer Science (CWI), Tech. Rep. INS-R9804;1–17, 1999, (last viewed Jul 2006). Available online: <http://ftp.cwi.nl/CWIREports/INS/INS-R9806.pdf>

- [99] I. Herman, G. Melançon, and M. Marshall, “Graph visualization and navigation in information visualization: A survey,” *IEEE Transactions on Visualization and Computer Graphics*. IEEE Computer Society, 2000, vol. 6, no. 1, pp. 24–43.
- [100] Human Genome Project Information. (last viewed Jul 2006). Available online: <http://www.doegenomes.org>
- [101] M. Honeyford, “Weighing in on Java native compilation: The pros and cons of generating native code from Java source,” developerWorks, IBM UK Labs, Tech. Rep., 2002, (8pps), (last viewed Jul 2006). Available online: <http://www-128.ibm.com/developerworks/java/library/j-native.html>
- [102] S.-H. Hong and N. S. Nikolov, “Layered drawings of directed graphs in three dimensions,” in *Asia-Pacific Symposium on Information Visualisation, APVIS 2005*, S.-H. Hong, Ed. Australian Computer Society, 2005, pp. 69–74.
- [103] S.-H. Hong and N. S. Nikolov, “Hierarchical layouts of directed graphs in three dimensions,” in *Graph Drawing: 13th International Symposium, GD 2005*, P. Healy and N. S. Nikolov, Eds. Springer, 2006, pp. 251–261.
- [104] K. Hornbæk, “Current practice in measuring usability: Challenges to usability studies and research,” *International Journal of Human-Computer Studies*. Academic Press, 2006, vol. 64, pp. 79–102.
- [105] K. Howard, “Special industry report: The Bioinformatics gold rush,” *Scientific American*. Scientific American, 2000, vol. 283, no. 1, pp. 58–63.
- [106] T. Hughes, Y. Hyun, and D. Liberles, “Visualising very large phylogenetic trees in three dimensional hyperbolic space,” *BMC Bioinformatics*. BioMed Central, 2004, vol. 5, no. 48, (6 pps).
- [107] A. Inselberg and B. Dimsdale, “Parallel coordinates: a tool for visualizing multi-dimensional geometry,” in *Proceedings of the First IEEE Conference on Visualization. Visualization '90*. IEEE Computer Society Press, 1990, pp. 361–378.
- [108] A. Inselberg, C. Grinstein, T. Mihalisin, and H. Hinterberger, “Visualizing multidimensional (multivariate) data and relations,” in *Proceedings, IEEE Conference on Visualization, 1994*. IEEE Computer Society Press, 1994, pp. 404–409.
- [109] A. Inselberg, “Visualization and data mining of high-dimensional data,” *Chemometrics and Intelligent Laboratory Systems*. Elsevier Science Publishers, 2002, vol. 60, no. 1-2, pp. 147–159.
- [110] D. Jerding and J. Stasko, “The information mural: a technique for displaying and navigating large information spaces,” *IEEE Transactions on Visualization and Computer Graphics*. IEEE Computer Society, 1998, vol. 4, no. 3, pp. 257–271.
- [111] G. Jimenez-Sanchez, B. Childs, and D. Valle, “Human disease genes,” *Nature*. Nature Publishing Group, 2001, vol. 409, pp. 853–855.
- [112] B. Johnson and B. Shneiderman, “Tree-maps: a space-filling approach to the visualization of hierarchical information structures,” in *Proceedings, IEEE Conference on Visualization, 1991*. IEEE Computer Society Press, 1991, pp. 284–291.

- [113] P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds., *Usability Evaluation in Industry*. Taylor & Francis, 1996, (224 pps).
- [114] D. Keim and H.-P. Kriegel, "Visualization techniques for mining large databases: a comparison," *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society, 1996, vol. 8, no. 6, pp. 923–938.
- [115] J. Kelso, N. Mulder, J. Visagie, and W. Hide, "Using ontologies in biological research," in *International Conference on Intelligent Systems for Molecular Biology / European Conference on Computational Biology (ISMB/ECCB 2004)*, 2004, pp. 52–57.
- [116] C. Keskin and V. Vogelmann, "Effective visualization of hierarchical graphs with the cityscape metaphor," in *NPIV '97: Proceedings of the 1997 Workshop on New Paradigms in Information Visualization and Manipulation*. ACM Press, 1997, pp. 52–57.
- [117] M. Kreuseler and H. Schumann, "A flexible approach for visual data mining," *IEEE Transactions on Visualization and Computer Graphics*,. IEEE Computer Society, 2002, vol. 8, no. 1, pp. 39–51.
- [118] H. P. Kumar, C. Plaisant, and B. Shneiderman, "Browsing hierarchical data with multi-level dynamic queries and pruning," *International Journal of Human-Computer Studies*. Academic Press, 1997, vol. 46, no. 1, pp. 103–124.
- [119] P. Lambrix, M. Habbouche, and M. Prez, "Evaluation of ontology development tools for bioinformatics," *Bioinformatics*. Oxford University Press, 2003, vol. 19, no. 12, pp. 1564–1571.
- [120] J. Lamping and R. Rao, "The hyperbolic browser: A focus+context technique for visualizing large hierarchies," *Journal of Visual Languages & Computing*. Academic Press, 1996, vol. 7, no. 1, pp. 33–55.
- [121] E. S. Lander *et al.*, "Initial sequencing and analysis of the human genome," *Nature*. Nature Publishing Group, 2001, vol. 409, pp. 860–921.
- [122] K.-L. Ma, "Large-scale data visualization," *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2001, vol. 21, no. 4, pp. 22–23.
- [123] J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy, "Representing and reasoning about mappings between domain models," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI 2002)*. AAAI Press, 2002, pp. 80–86.
- [124] V. Markowitz and T. Topaloglou, "Applying data warehouse concepts to gene expression data management," in *2nd International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*. IEEE Computer Society, 2001, pp. 65–72.
- [125] M. Mascoet, "Interaction and visualization supporting Web browsing patterns," in *Proceedings, Fifth International Conference on Information Visualisation*, E. Banissi, F. Khosrowshahi, M. Sarfraz, and A. Ursyn, Eds. IEEE Computer Society, 2001, pp. 413–418.
- [126] N. Mays and C. Pope, "Qualitative research in health care. assessing quality in qualitative research," *British Medical Journal*. BMJ Publishing Group Ltd, 2000, vol. 320(7226), pp. 50–52.

- [127] T. Munzner, “H3: laying out large directed graphs in 3D hyperbolic space,” in *Proceedings of VIZ '97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, J. Dill and N. Gershon, Eds. IEEE Computer Society Press, 1997, pp. 2–10, 114.
- [128] T. Munzner, “Exploring large graphs in 3D hyperbolic space,” *IEEE Computer Graphics and Applications*. IEEE Computer Society, 1998, vol. 18, no. 4, pp. 18–23.
- [129] T. Munzner, “Guest editor’s introduction — information visualization,” *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2002, vol. 22, no. 1, pp. 20–21.
- [130] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, “TreeJuxtaposer: scalable tree comparison using focus+context with guaranteed visibility,” *ACM Transactions on Graphics*. ACM Press, 2003, vol. 22, no. 3, pp. 453–462.
- [131] E. J. Nestler and D. Landsman, “Learning about addiction from the genome,” *Nature*. Nature Publishing Group, 2001, vol. 409, pp. 834–835.
- [132] C. Nevill-Manning, “The biological digital library,” *Communications of the ACM*. ACM Press, 2001, vol. 44, no. 5, pp. 41–42.
- [133] J. Nielsen, *Usability Engineering*. Academic Press, 1994, (362 pps).
- [134] N. F. Noy and M. A. Musen, “The PROMPT suite: Interactive tools for ontology merging and mapping,” *International Journal of Human-Computer Studies*. Academic Press, 2003, vol. 59, no. 6, pp. 983–1024.
- [135] N. F. Noy, R. W. Ferguson, and M. A. Musen, “The knowledge model of Protégé-2000: Combining interoperability and flexibility,” in *Knowledge Acquisition, Modeling and Management, 12th International Conference, EKAW 2000*, ser. Lecture Notes in Computer Science, R. Dieng and O. Corby, Eds. Springer, 2000, pp. 17–32.
- [136] H. Parkinson, S. Aitken, R. A. Baldock, J. B. L. Bard, A. Burger, T. F. Hayamizu, A. Rector, M. Ringwald, J. Rogers, C. Rosse, C. J. Stoeckert Jr., and D. Davidson, “The SOFG anatomy entry list (SAEL): an annotation tool for functional genomics data,” *Comparative and Functional Genomics*. John Wiley & Sons, 2004, vol. 5, pp. 521–527.
- [137] N. W. Paton, S. A. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, C. A. Goble, S. J. Hubbard, and S. G. Oliver, “Conceptual modelling of genomic information,” *Bioinformatics*. Oxford University Press, 2000, vol. 16, no. 6, pp. 548–557.
- [138] K. Perlin and D. Fox, “Pad: an alternative approach to the computer interface,” in *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer Graphics and Interactive Techniques*. ACM Press, 1993, pp. 57–64.
- [139] C. Plaisant, J. Grosjean, and B. Bederson, “SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation,” in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, P. C. Wong and K. Andrews, Eds. IEEE Computer Society, 2002, pp. 57–64.
- [140] S. Pook, G. Vaysseix, and E. Barillot, “Zomit: biological data visualization and browsing,” *Bioinformatics*. Oxford University Press, 1998, vol. 14, no. 9, pp. 807–814.

- [141] J. Preece, *A Guide to Usability: Human Factors in Computing*. Addison-Wesley, 1993, (144 pps).
- [142] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*. Addison-Wesley, 1994, (816 pps).
- [143] H. Purchase, "The effects of graph layout," in *Proceedings, 1998 Australasian Computer Human Interaction Conference (OzCHI'98)*, P. Calder and B. Thomas, Eds. IEEE Computer Society, 1998, pp. 80–86.
- [144] J. Quackenbush, "Computational genetics: Computational analysis of microarray data," *Nature Reviews Genetics*, 2001, vol. 2, no. 6, pp. 418–427.
- [145] J. Quackenbush, "The power of public access: the human genome project and the scientific process," *Nature Genetics*. Nature Publishing Group, 2001, vol. 29, no. 1, pp. 4–6.
- [146] A. Rector and J. Rogers, "Ontological & practical issues in using a description logic to represent medical concepts: Experience from GALEN," University of Manchester School of Computer Science, Tech. Rep. Preprints CSPP-35;:1–33, 2005, (last viewed Jul 2006). Available online: <http://www.cs.manchester.ac.uk/cspreprints/PrePrints/cspp35.pdf>
- [147] J. Rekimoto and M. Green, "The information cube: Using transparency in 3D information visualization," in *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, 1993, pp. 125–132.
- [148] P. Rheingans, "Are we there yet? Exploring with dynamic visualization," *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2002, vol. 22, no. 1, pp. 6–10.
- [149] J. E. Richardson, J. A. Kadin, J. A. Blake, C. J. Bult, J. T. Eppig, and M. Ringwald, "From sipping on a straw to drinking from a fire hose: Data integration in a public genome database," in *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004*. IEEE Computer Society 2004, 2004, pp. 795–799.
- [150] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich, "Data mountain: using spatial memory for document management," in *UIST '98: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, 1998, pp. 153–162.
- [151] G. G. Robertson, S. K. Card, and J. D. Mackinlay, "Information visualization using 3D interactive animation," *Communications of the ACM*. ACM Press, 1993, vol. 36, no. 4, pp. 57–71.
- [152] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: animated 3D visualizations of hierarchical information," in *CHI '91: Proceedings of the SIGCHI conference on Human Factors in Computing Systems: Reaching through Technology*, S. P. Robertson, G. M. Olson, and J. S. Olson, Eds. ACM Press, 1991, pp. 189–194.
- [153] A. J. Robinson and T. P. Flores, "Poster: Visualisation support at the European Bioinformatics Institute," <http://industry.ebi.ac.uk/~Ealan/Posters/VisSupport.ps.gz>, 1997, (last viewed Jul 2006).
- [154] A. J. Robinson and T. P. Flores, "Novel techniques for visualizing biological information," in *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*,

- T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander, and A. Valencia, Eds. AAAI Press, 1997, pp. 241–249.
- [155] C. Rosse, A. Kumar, J. L. Mejino Jr., D. L. Cook, L. T. Detwiler, and B. Smith, “A strategy for improving and integrating biomedical ontologies,” in *Proceedings, AMIA 2005 Annual Symposium, Biomedical and Health Informatics: From Foundations, to Applications to Policy*. AMIA, 2005, pp. 639–643.
- [156] C. Rosse and J. L. V. Mejino Jr., “A reference ontology for biomedical informatics: the Foundational Model of Anatomy,” *Journal of Biomedical Informatics*. Academic Press, 2003, vol. 36, no. 6, pp. 478–500.
- [157] J. Rubin, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., 1994, (352 pps).
- [158] C. Sander, “Bioinformatics - challenges in 2001,” *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 1, pp. 1–2.
- [159] M. Sarkar and M. H. Brown, “Graphical fisheye views of graphs,” in *CHI '92: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, P. Bauersfeld, J. Bennett, and G. Lynch, Eds. ACM Press, 1992, pp. 83–91.
- [160] M. Schroeder, D. Gilbert, J. van Helden, and P. Noy, “Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer,” *Information Sciences*. Elsevier Science, 2001, vol. 139, no. 1–2, pp. 19–57.
- [161] M. M. Sebrechts, J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller, “Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces,” in *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*. ACM Press, 1999, pp. 3–10.
- [162] B. Shneiderman, “Dynamic queries for visual information seeking,” *IEEE Software*. IEEE Computer Society, 1994, vol. 11, no. 6, pp. 70–77.
- [163] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” in *Proceedings, IEEE Symposium on Visual Languages, 1996*. IEEE Computer Society, 1996, pp. 336–343.
- [164] B. Shneiderman, “Why not make interfaces better than 3D reality?” *IEEE Computer Graphics and Applications*. IEEE Computer Society, 2003, vol. 23, no. 6, pp. 12–15.
- [165] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd ed. Addison-Wesley, 1998, (650 pps).
- [166] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 4th ed. Addison-Wesley, 2004, (672 pps).
- [167] L. D. Stein, “How Perl saved the Human Genome Project,” *The Perl Journal*, 1996, vol. 1, no. 2, (7 pps) (last viewed Jul 2006). Available online: http://www.bioperl.org/wiki/How_Perl_saved_human_genome
- [168] R. Stevens, C. Goble, and S. Bechhofer, “Ontology-based knowledge representation for bioinformatics,” *Briefings in Bioinformatics*. Oxford University Press, 2000, vol. 1, no. 4, pp. 398–414.

- [169] R. Stevens, C. Goble, P. Baker, and A. Brass, "A classification of tasks in bioinformatics," *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 2, pp. 180–188.
- [170] M. C. Stone, K. Fishkin, and E. A. Bier, "The movable filter as a user interface tool," in *CHI '94: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, B. Adelson, S. Dumais, and J. Olson, Eds. ACM Press, 1994, pp. 306–312.
- [171] M.-A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy, "Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé," in *Workshop on Interactive Tools for Knowledge Capture (K-CAP 2001)*, 2001, (9 pps).
- [172] M.-A. Storey and H. Müller, "Graph layout adjustment strategies," in *Proceedings, Symposium on Graph Drawing (GD 1995)*, F.-J. Brandenburg, Ed. Springer Verlag, 1995, pp. 487–499.
- [173] K. Sugiyama and K. Misue, "Visualization of structural information: automatic drawing of compound digraphs," *IEEE Transactions on Systems, Man and Cybernetics*. IEEE, 1991, vol. 21, no. 4, pp. 876–892.
- [174] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," *IEEE Transactions on Systems, Man, and Cybernetics*. IEEE, 1981, vol. 11, no. 2, pp. 109–125.
- [175] A. Sutcliffe, "On the effective use and reuse of HCI knowledge," *ACM Transactions on Computer-Human Interaction*. ACM Press, 2000, vol. 7, no. 2, pp. 197–221.
- [176] R. Tamassia, G. Di Battista, and C. Batini, "Automatic graph drawing and readability of diagrams," *IEEE Transactions on Systems, Man and Cybernetics*. IEEE, 1988, vol. 18, no. 1, pp. 61–79.
- [177] M. Tory, A. Kirkpatrick, M. Atkins, and T. Moller, "Visualization task performance with 2D, 3D, and combination displays," *IEEE Transactions on Visualization and Computer Graphics*. IEEE Computer Society, 2006, vol. 12, no. 1, pp. 2–13.
- [178] E. R. Tufte, *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997, (151 pps).
- [179] E. Tufte, *Envisioning Information*. Graphics Press, 1990, (126 pps).
- [180] R. Turner, K. Chaturvedi, N. Edwards, D. Fasulo, A. Halpern, D. Huson, O. Kohlbacher, J. Miller, K. Reinert, K. Remington, R. Schwartz, B. Walenz, S. Yooseph, and S. Istrail, "Visualization challenges for a new cyber-pharmaceutical computing paradigm," in *Proceedings, IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*. IEEE, 2001, pp. 7–18, 145.
- [181] M. Velez, D. Silver, and M. Tremaine, "Understanding visualization through spatial ability differences," in *IEEE VIS 05, Proceedings of Visualization 2005*, C. T. Silva, E. Gröller, and H. Rushmeier, Eds. IEEE Computer Society, 2005, pp. 511–518.
- [182] N. G. Vinson, "Design guidelines for landmarks to support navigation in virtual environments," in *CHI '99: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM Press, 1999, pp. 278–285.
- [183] S. Wilson and J. Kesselman, *Java Platform Performance: Strategies and Tactics*, 1st ed. Addison-Wesley, 2000, (230 pps).

- [184] P. Young, “Three dimensional information visualisation,” Visualisation Research Group, Centre for Software Maintenance, Department of Computer Science, University of Durham, Tech. Rep. 12/96;:1–37, 1996, (last viewed Jul 2006). Available online: <http://www.dsi.unive.it/~smm/2000/docs/iv/iv-survey.html>
- [185] E. M. Zdobnov, R. Lopez, R. Apweiler, and T. Etzold, “The EBI SRS server — recent developments,” *Bioinformatics*. Oxford University Press, 2002, vol. 18, no. 2, pp. 368–373.
- [186] C. M. Zmasek and S. R. Eddy, “ATV: display and manipulation of annotated phylogenetic trees,” *Bioinformatics*. Oxford University Press, 2001, vol. 17, no. 4, pp. 383–384.